

AÑO II N.º 47

**MICROHOBBY**

# AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

*Semanal*

AÑO II N.º 47

**160 Ptas.**

Canarias 165 pts.

LA BATALLA  
DEL SIGLO EN  
ESTRATOREACTOR  
ES SKY-REED

PROGRAMACION  
ESTRUCTURADA:  
POR EL CAOS  
HACIA EL ORDEN

LAS VENTANAS  
DE TU AMSTRAD  
EN CODIGO  
MAQUINA.

No más problemas  
con las estructuras  
de datos.  
¡Lo contamos  
todo!

**SOFTWARE**

COMANDO,  
VENCER O MORIR  
¿ALCANZARAS  
LA ULTIMA  
POSICION  
ENEMIGA?



HOBBY PRESS



SOMOS MAYORISTAS

# MICRO-1

PRECIOS  
INCLUIDO IVA

C/ Duque de Sesto, 50. 28009 Madrid  
Tel.: (91) 274 53 80

(Metro O'Donnell o Goya)  
Aparcamiento gratuito en Felipe II

Ofertas en software: 2 programas al precio de 1 y además regalo fin de curso una calculadora completamente gratis. ¡¡Asombroso!!  
¿Verdad?

BAT MAN	2.300	ptas.	KUNG-FU MASTER	2.300	ptas.
ROCK'N LUCHA	2.300	ptas.	SABOTEUR	2.300	ptas.
YIER AR KUNG FU	2.300	ptas.	PING PONG	2.300	ptas.
THE WAY OF THE TIGER	2.300	ptas.	MILLION II	0000	ptas.
WEST BANK	2.100	ptas.	OLE TORO	2.300	ptas.
CAMELOT WARRIORS	2.300	ptas.	TURBO ESPRIT	2.100	ptas.
RAMBO	2.300	ptas.	SABRE WULF	1.650	ptas.
WORLD CUP (DISCO)	3.300	ptas.	BATALLA DE LOS PLANETAS	2.100	ptas.
MILLION II (DISCO)	3.300	ptas.	SABOTEUR-COMBAT LINX DISCO	3.300	ptas.

RAMBO-MATCH DAY (DISCO) 3.300 ptas.

SOFTWARE DE REGALO (OFERTA 2x1)  
DECATHLON BEACH HEAD SOUTHERN BELLE  
DRAGONTORC

LAPIZ OPTICO  
3.295 PTAS.

CASSETTE ESPECIAL ORDENADOR  
5.295 PTAS.

SINTETIZADOR DE VOZ EN  
CASTELLANO  
7.650 PTAS.

AMPLIACION DE MEMORIA ANTA 64 K.3  
12.500 PTAS.

**IMPRESORAS**  
**20% DE DESCUENTO SOBRE P.V.P.**

TAPA DE METACRILATO 464	895
CABLE CENTRONICS	3.175
CABLE SEGUNDA UNIDAD D.	1.790
CABLE SEPARADORES 6128	1.975
INTERFACE RS232	9.265
CINTA VIRGEN C15	69

CABLE AUDIO	795
CABLE ADAPTADOR 2 JOYSTICK	2.390
CABLE SEPARADORES 464	1.390
CABLE SEPARADOR 8256	2.900
CABLE RS232	2.500
DISKETTES 3"	990

**PRECIOS EXCEPCIONALES PARA TU AMSTRAD**  
**CPC-464, CPC-6128, PCW-8256**

¡¡LLEGARON LAS REBAJAS DE VERANO A MICRO 1!!

OFERTAS EN JOYSTICKS	
QUICK SHOT I	1.395
QUICK SHOT II	1.695
QUICK SHOT V	1.695

**PEDIDOS CONTRA REEMBOLSO SIN NINGUN GASTO DE ENVIO TEL. (91) 274 53 80**  
**O ESCRIBIENDO A: MICRO-1. C/ DUQUE DE SESTO, 50. 28009 MADRID.**

Tiendas y distribuidores grandes descuentos.  
Dirigirse a Diproimsa. C/ Galatea, 25. Tel. (91) 742 20 19 ó 742 79 68



**Director Editorial**  
José I. Gómez-Centurión

**Director Ejecutivo**

José M.ª Díaz

**Redactor Jefe**

Juan José Martínez

**Diseño gráfico**

Fernando Chaumel

**Colaboradores**

Eduardo Ruiz

Javier Barceló

David Sopuerta

Robert Chatwin

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalia Gómez

Alberto Suñer

**Secretaría Redacción**

Carmen Santamaría

**Fotografía**

Carlos Candel

Chema Sacristán

**Portada**

M. Barco

**Ilustradores**

J. Igual, J. Pons, F. L. Frontán,  
J. Septien, Pejo, J. J. Mora

**Edita**

HOBBY PRESS, S.A.

**Presidente**

Mario Andriño

**Consejero Delegado**

José I. Gómez-Centurión

**Jefe de Producción**

Carlos Peropadre

**Marketing**

Marta García

**Jefe de Publicidad**

Concha Gutiérrez

**Publicidad Barcelona**

José Galán Cortés

Tel: (93) 303 10 22/313 71 62

**Secretaría de Dirección**

Marisa Cogorro

**Suscripciones**

M.ª Rosa González

M.ª del Mar Calzada

**Redacción, Administración  
y Publicidad**

Ctra. de Irún km 12,400

(Fuencarral) 28049 Madrid

Teléfonos: Suscrip.: 734 65 00

Redacción: 734 70 12

**Dto. Circulación**

Paulino Blanco

**Distribución**

Coedis, S. A. Valencia, 245

Barcelona

**Imprime**

ROTECIC, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

**Fotocomposición**

Novocomp, S.A.

Nicolás Morales, 38-40

**Fotomecánica**

GROF

Ezequiel Solana, 16

**Depósito Legal:**

M-28468-1985

Derechos exclusivos  
de la revista

**COMPUTING with  
the AMSTRAD**

Representante para Argentina, Chile,  
Uruguay y Paraguay, Cia.  
Americana de Ediciones, S.R.L. Sud  
America 1.532. Tel.: 21 24 64. 1209  
BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace  
necesariamente solidario de las  
opiniones vertidas por sus  
colaboradores en los artículos  
firmados. Reservados todos los  
derechos.

**MICROHOBBY**

# AMSTRAD

## sumario

Año II • Número 47 • 22 al 28 de Julio de 1986  
160 ptas. (incluido I.V.A.)  
Canarias, 155 ptas. + 10 ptas. sobretasa aérea.  
Ceuta y Melilla, 155 ptas.

### Primeros Pasos

6

Semana a semana descubrimos nuevas sutilezas del LOCOMOTIVE de AMSTRAD. En esta ocasión nos ocupamos de un tema donde vuelve a brillar la potencia de este Basic que no deja de sorprendernos: se trata del tratamiento de cadenas o funciones de librería, que dirían los entendidos. La fragmentación de cadenas alfanuméricas es el primer tema. Estamos convencidos de que disfrutaréis en esta nueva parcela del Basic.



### 12 Programación estructurada

Los hay que piensan que programar es sentarse delante de un ordenador y ponerse a teclear como una fiera. No pretendemos cambiar las ideas a nadie, cada cual en su casa que barra como quiera. Sin embargo, hablando objetivamente, un programa estructurado y estudiado previamente a conciencia, tiene muchas más posibilidades de éxito y corrección que cualquier otro que no haya sido diseñado de esta forma. Primero, antes de elegir tu método, lee este artículo: después decide.

18

### Código Máquina

Hace ya muchos números, tratamos cómo utilizar las ventanas en el Amstrad; ahora volvemos de nuevo sobre el tema, pero en esta ocasión lo analizamos bajo la siempre veloz perspectiva del código máquina.

### Serie Oro

22

La invasión se ha hecho realidad; bombarderos y cazas se han lanzado en ataque continuo contra tu base. Intenta contrarrestar la invasión con tus misiles tierra-aire, teledirigidos y convencionales, y que el cielo se tiña de rojo.



### ProgramAcción

26

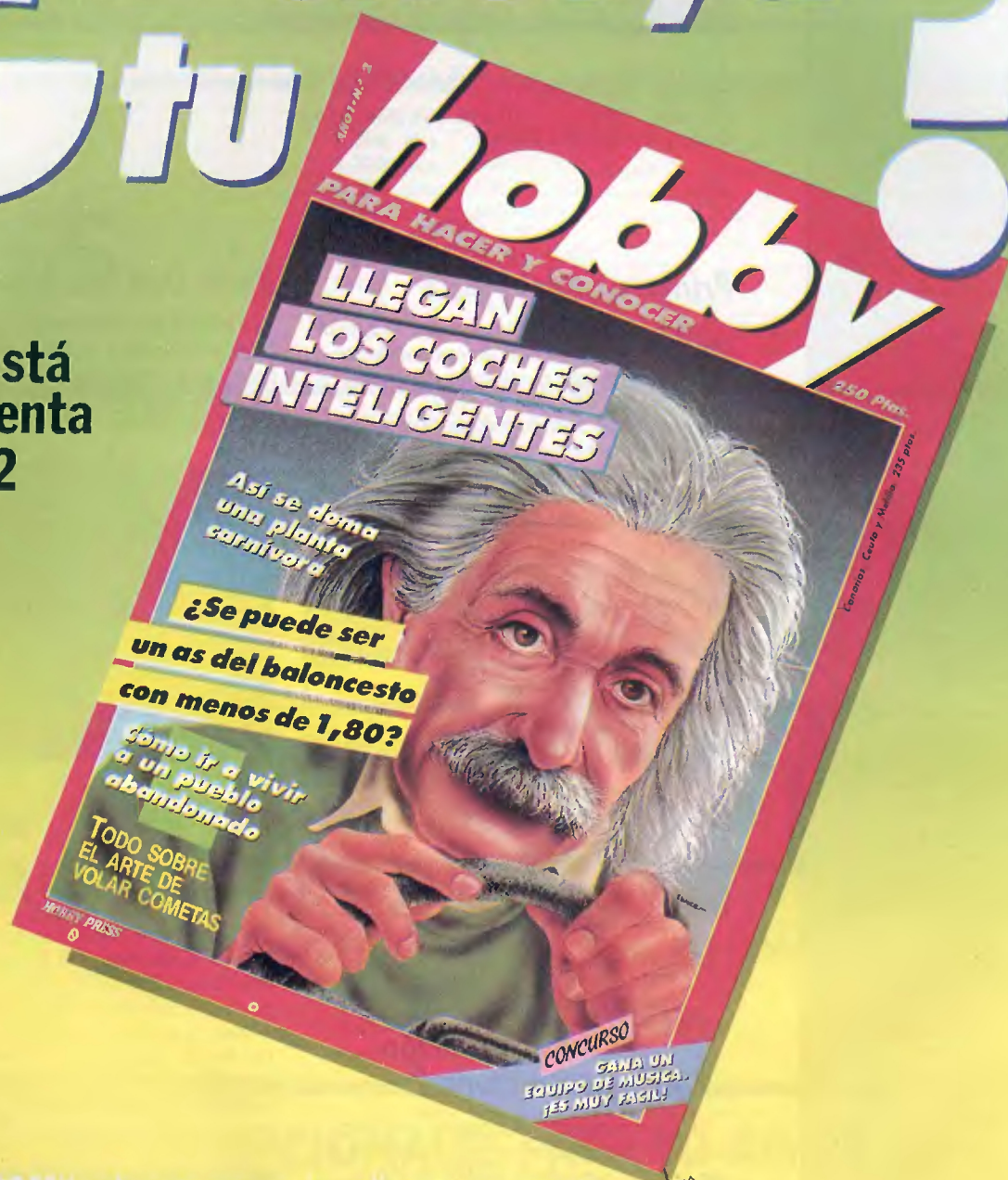
Si algún defecto hay que encontrarle al Basic, éste sería, sin duda, la ausencia de estructuras de datos diseñados por el propio usuario. En este artículo estudiamos cómo con un poco de imaginación y método, es posible simular este tipo de estructuras que acercan el programa de forma racional, un poco más a lenguajes más evolucionados. Comenzamos estudiando las pilas, una de las estructuras más utilizadas y versátiles dentro de la programación.



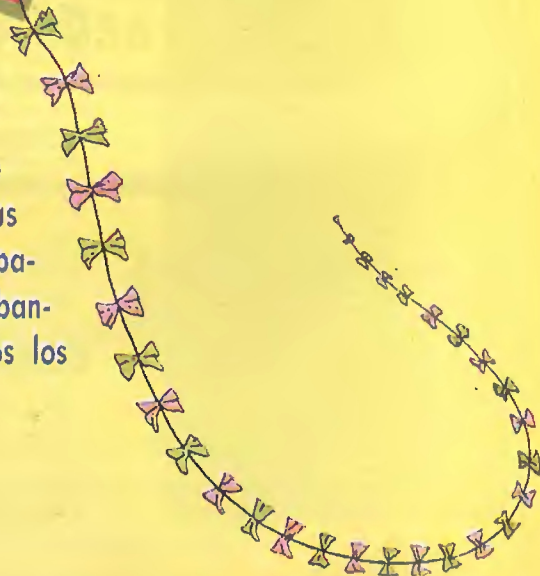


# ¿Tienes ya? tu **hobby**

Ya está  
a la venta  
el n.º 2



HOBBY sale este mes a la calle con un montón de temas insólitos que seguro te apasionarán. ¿Sabías, por ejemplo, que puedes ser un campeón de baloncesto aunque midas menos de 1,80? ¿O que es posible domar una planta carnívora y medir sus estímulos? HOBBY te enseña también cosas tan disparates como los pasos a seguir para irte a un pueblo abandonado o el arte de volar cometas. Y, además, todos los meses regalamos una cadena de música. Ten un HOBBY todos los meses. ¡Ya está en tu kiosco el n.º 2!





## DOS NUEVOS PERIFERICOS PARA TU AMSTRAD



**I**NDESCOMP nos informa del lanzamiento de un nuevo joystick y de una nueva impresora para los **Amstrad CPC**.

El **AMSTICK**, así se llama el joystick, es de sólida construcción y suave manejo. Va provisto de botón de disparo en la palanca de mando, así como en la base que dispone de potentes ventosas que le darán una segura sujeción.

Este joystick es compatible con los ordenadores Sinclair, Atari, Commodore, etc.

Y si bien esto es interesante, lo que le convierte en verdaderamente apetecible en su precio, 1.000 ptas. (IVA incluido) que le harán super competitivo en el mercado:

Como de super competitiva se puede catalogar la **DMP-2000**, impresora que viene a ser un peldaño más arriba en la gama de las disponibles para los **CPC**.

Se puede conectar a cualquier otro ordenador que tenga interface paralelo Centronics, y es compatible con el standar **EPSON**.

Su sistema de impresión es de punto a punto en diferentes densidades, dispone de tipos de letra estándar, élite, cursiva, alta calidad, etc.

La velocidad de impresión es de 105 caracteres por segundo. Admite papel continuo de 114 mm a 254 mm de ancho, así como hojas sueltas de 102 mm a 241 mm de ancho.

La carga del papel es frontal y está dotada de patas abatibles que permiten colocar papel debajo de ella para ahorrar espacio.

PINBALL  
WIZARD

Este juego nació originariamente para Spectrum cinta, y ahora ya tenemos en el mercado la versión disco.

Con este juego podemos vivir una divertida partida de flipper, con la particularidad interesante de poder jugar a cuatro mandos. Al igual que en las máquinas de los bares, el Pinball Wizard cuenta con puntuaciones

Primera  
PLANA

especiales, bonos e incluso bola extra, todo ello con la ventaja de ahorrarse los cinco dures de la partida.

Este entretenido juego pertenece a la casa **ACE Software** y es distribuido por **MICROBYTE**.

## AMSTRAD AL VOLANTE: EL EQUIPO AMSTRAD EN CABEZA DEL CAMPEONATO

**E**l equipo **AMSTRAD** participante en el Campeonato de España de Producción, a través de su piloto «Correcaminos» se encuentra actualmente en primera posición del Campeonato de Velocidad en Circuitos, también denominado de Producción.

Tras cuatro pruebas celebradas en Jerez, Jarama, Toledo y Puerto Banús, las clasificaciones obtenidas (tres segundos puestos y un primero) dan idea de la profesionalidad de todo el equipo, que bajo la dirección de **AMSTRAD** y la preparación técnica

de Cecilio Muñiz, ha hecho posible poder afrontar la segunda parte del campeonato con una cierta tranquilidad cara al resultado final.

El Renault 11 Turbo, que en un principio parecía no poder inquietar a sus más directos rivales, los Volkswagen Golf GTI preparados por **RAS**, se ha convertido en el «coco» de los potentes coches alemanes, siendo en este momento el único y más peligroso rival que encuentran en los circuitos.

En la próxima carrera, Calafat, el coche del Equipo de Competición **AMSTRAD** España, estrenará nuevas suspensiones con las que esperamos pueda poner aún más tierra por medio entre sus seguidores.





# LA FRAGMENTACION ESTA SERVIDA

*Antaño vimos que todo lo que hay en el interior de un ordenador, o mejor dicho en su memoria, es una colección de «ceros» y «unos». Conociendo un poquito el sistema binario de numeración, pudimos pensar que si cogemos un grupo de ellos y lo tratamos convenientemente obtendríamos el equivalente decimal del número binario que habíamos formado. Y así es.*



ero nuestro **Amstrad** no se queda ahí. También es capaz de transformar estos conjuntos de «bits» activados (iguales a 1) o desactivados (valiendo 0) en unos códigos que se corresponden con todos los caracteres del alfabeto, signos especiales —por ejemplo el «+»— e incluso con símbolos gráficos. es lo que llamamos «**código Ascii**».

Agrupando estos «**caracteres**» formamos lo que se ha llamado «**cadenas**» de caracteres o «**strings**». Si son constantes (permanecen siempre igual a lo largo de un programa) las denominaremos «**constantes alfanuméricas**», y podremos reconocerlas donde veamos una serie de caracteres encerrados entre comillas, como por ejemplo:

**"PEPE"**

Cuando, por el contrario, su valor se cambia con alguna instrucción Basic dentro de un programa, lo suyo es definir una variable de este tipo en la que se vayan almacenando los grupos de caracteres que en cada momento les vayamos dando. ¿Recuerda cómo se definían dichas variables?

Con:

nombre\$

estamos refiriendonos a un espacio de la memoria del ordenador que vamos a reservar para guardar en él valores o datos alfanuméricos (*cadena de caracteres*). Cada vez que nos refiramos a ellos, podremos hacerlo con el nombre que le hayamos asignado, como en este caso sería «**nombres**».

El signo «\$» que hemos colocado detrás del nombre dado a la variable alfanumérica es una especie de etiqueta, con la que le estamos indicando al ordenador que se trata de una cadena de caracteres y no un número.

Para dar un determinado valor a esta variable bastaría con poner una instrucción tal como:

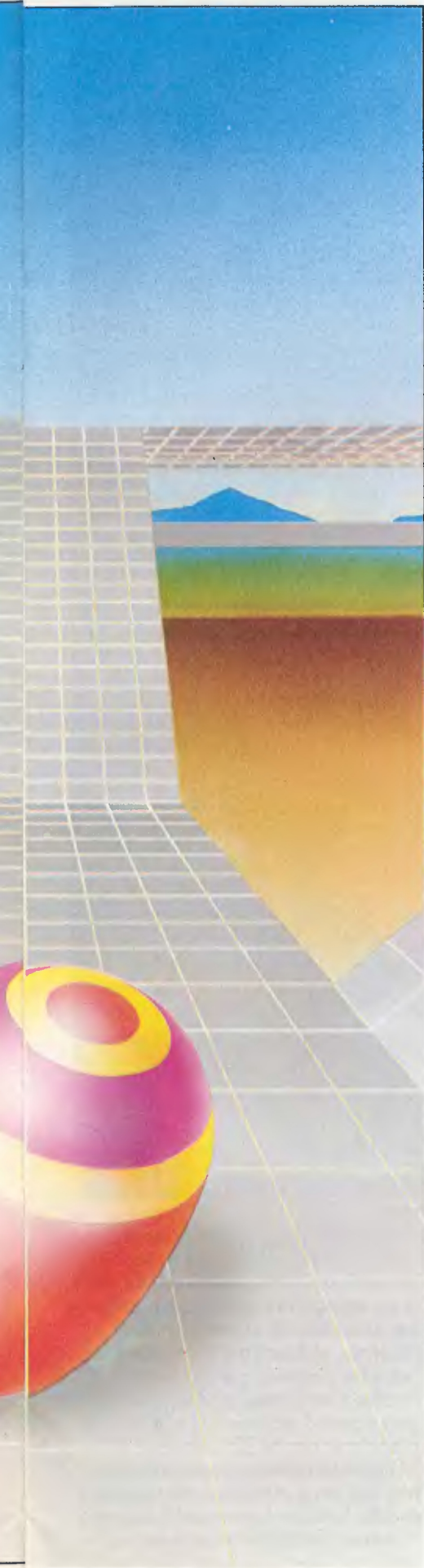
nombre\$="PEPE"

y en la zona de memoria reservada bajo ese nombre almacenaríamos el valor «**literal**» que está entre comillas, o sea, PEPE.

Con todo esto lo que pretendemos es refrescarle un poco sus conocimientos, si es menester, sobre este tipo de variables e intentar convencerle de que dentro de un ordenador no sólo se pueden almacenar números como tales, sino también series de ceros y unos (o códigos) que se corresponden con todos y cada uno de los caracteres que están sobre el teclado además de un «**montón**» de símbolos gráficos que le vendrán muy bien para sus propias aplicaciones.

El **Amstrad** tiene gran cantidad de sentencias o funciones que vamos a poder aplicar sobre este tipo de variables. ¡No se ha quedado corto!. Podemos decir que con ellas será posible hacer casi todo lo que se le está pasando por la cabeza en este momento a cualquier lector inquieto o atrevido.





Bueno, vayamos al grano. En uno de los artículos de esta sección vimos ya una serie de ellas que suponemos no habrá olvidado. Como iniciativa, vale, pero no es nuestra intención quedarnos detenidos ahí. Ha llegado el momento de ampliar todo lo que ya sabía sobre «cadenas», si es posible, viendo alguna que otra nueva función relacionada con este campo tal como le prometimos en su día.

Lo primero que haremos será intentar «fragmentar» o descomponer una de estas cadenas. Si es la primera vez que escucha esta palabra puede ocurrir que le suene a algo parecido a «romper» o estropear, pero no es así. Es bastante más útil.

Significa que vamos a seleccionar una parte de una cadena para hacer cualquier cosa con ella, informáticamente hablando, claro. Escogeremos una o varias letras y se las asignaremos a otra variable, las visualizaremos en la pantalla o realizaremos un complicado cálculo con ellas.

Vamos a descomponer su nombre, por ejemplo, en diferentes grupos de letras. ¿Por dónde empezamos? Imaginamos que le parecerá muy descabellado el hecho de que hallamos decidido comenzar por la primera letra.

A partir de ella iremos formando conjuntos de una, dos, tres, etc..., letras y las escribiremos en la pantalla. Escriba el programa 1 y ejecútelo. A continuación, analizaremos los resultados obtenidos.

### Programa uno

Es muy sencillo, ya que se trata simplemente de un bucle FOR... NEXT cuya variable de control «índice» va tomando valores comprendidos entre 1 y la longitud del nombre que haya tecleado.

Para calcular el número de letras de su nombre empleamos, en la línea 50, una función que no le será desconocida:

LEN(nombre\$)

que nos devuelve, cuando la invocamos, la longitud de la cadena que hay entre los paréntesis. ¿Recuerda?

Para hacer la selección, o «fragmentación», de las letras que deseamos utilizamos, como se ve en la línea 70, una nueva función:

LEFT\$(nombre\$, número)

descompone la cadena «nombre\$» en sus distintas letras y nos da un valor literal compuesto de tantas como indica «número» tomándolas a partir de la izquierda.

Este valor, que también es una cadena, es el que asignamos a «fragmento\$» en la línea 70 para después imprimirla.

En el programa delimitamos la longitud del bucle. Además de servirnos para controlar el número de veces que repetiremos el cuerpo del mismo, también lo podemos emplear para multitud de cosas.

## Primeros PASOS

Escriba:

fragmento\$ = LEFT\$("ABCDE", 3)

y a continuación:

PRINT fragmento\$

¿Qué cadena aparecerá en la pantalla? Analizando paso a paso estas dos últimas órdenes podrá contestar a esta pregunta. En la primera asignamos a la variable «fragmento\$» tantos caracteres del literal «ABCDE» como nos indique el segundo miembro del paréntesis, o sea, 3.

¿A partir de dónde comenzamos a contar? Según la definición de LEFT\$ lo tenemos muy claro: por la izquierda. Luego la segunda orden nos visualizará el fragmento:

ABC

¿Comprendido?

En nuestro programa el valor del segundo «parámetro» de la función —su longitud— es el que va tomando la variable de control del bucle, «índice», en cada vuelta del mismo. Irá incrementándose desde uno, lo que dará como resultado una subcadena de una sola letra, hasta que alcance la longitud total de su nombre calculada con LEN.

¿Qué pasaría si este segundo parámetro supera el número de letras del literal sobre el que aplicamos la función? Compruébelo tecleando:

fragmento\$ = ("ABC", 6)

```
10 REM PROGRAMA 1
20 CLS
30 INPUT "¿CUAL ES SU NOMBRE? ", nombre$
40 PRINT
50 longitud=LEN(nombre$)
60 FOR indice=1 TO longitud
70 fragmento$=LEFT$(nombre$, indice)
80 PRINT fragmento$
90 NEXT indice
100 PRINT
```

Ahora estamos intentando asignar 6 caracteres tomados a partir de la izquierda de la palabra. ¿Cómo será posible hacerlo si sólo tiene 3? Pues bien, al Amstrad no le engañamos. Toma solamente los tres caracteres y punto. Podemos comprobarlo escribiendo:

PRINT fragmento\$

y

PRINT LEN (fragmento\$)

visualizándose en la pantalla las tres letras de



«ABC» y además la longitud del fragmento (tres) que nos confirma lo que acabamos de decir. Podemos resumirlo asegurando que si la longitud de la expresión literal es menor que la longitud deseada, la cadena que produce nuestro ordenador es la expresión completa.

Para no ser menos, también existe otra función que nos va a seleccionar una serie de letras, pero ahora a partir de la palabra o dato alfanumérico que queramos. Se trata de:

```
RIGHT$(cadena$,longitud)
```

y nos produce una subcadena que tendrá tantos caracteres como nos indique «longitud» tomados por la derecha del valor alfanumérico contenido en «**cadena**».

```
PRINT RIGHT$("PEPE",3)
```

hará que se escriban 3 caracteres de la palabra «**PEPE**» empezando a contar por la derecha de la misma. O sea: «**EPE**»

```
PRINT RIGHT$("12345",2)
```

¿qué efectos producirá? Intente explicárselo por sí mismo y compárelos con esta otra instrucción:

```
PRINT LEFT$("12345",2)
```

Tanto la expresión literal como la longitud de la subcadena elegida son las mismas, pero las funciones son diferentes. ¿Ve claramente el porqué de los diferentes resultados obtenidos?

Con el programa 2, que es muy parecido al anterior, vemos una forma práctica de aplicar esta nueva función.

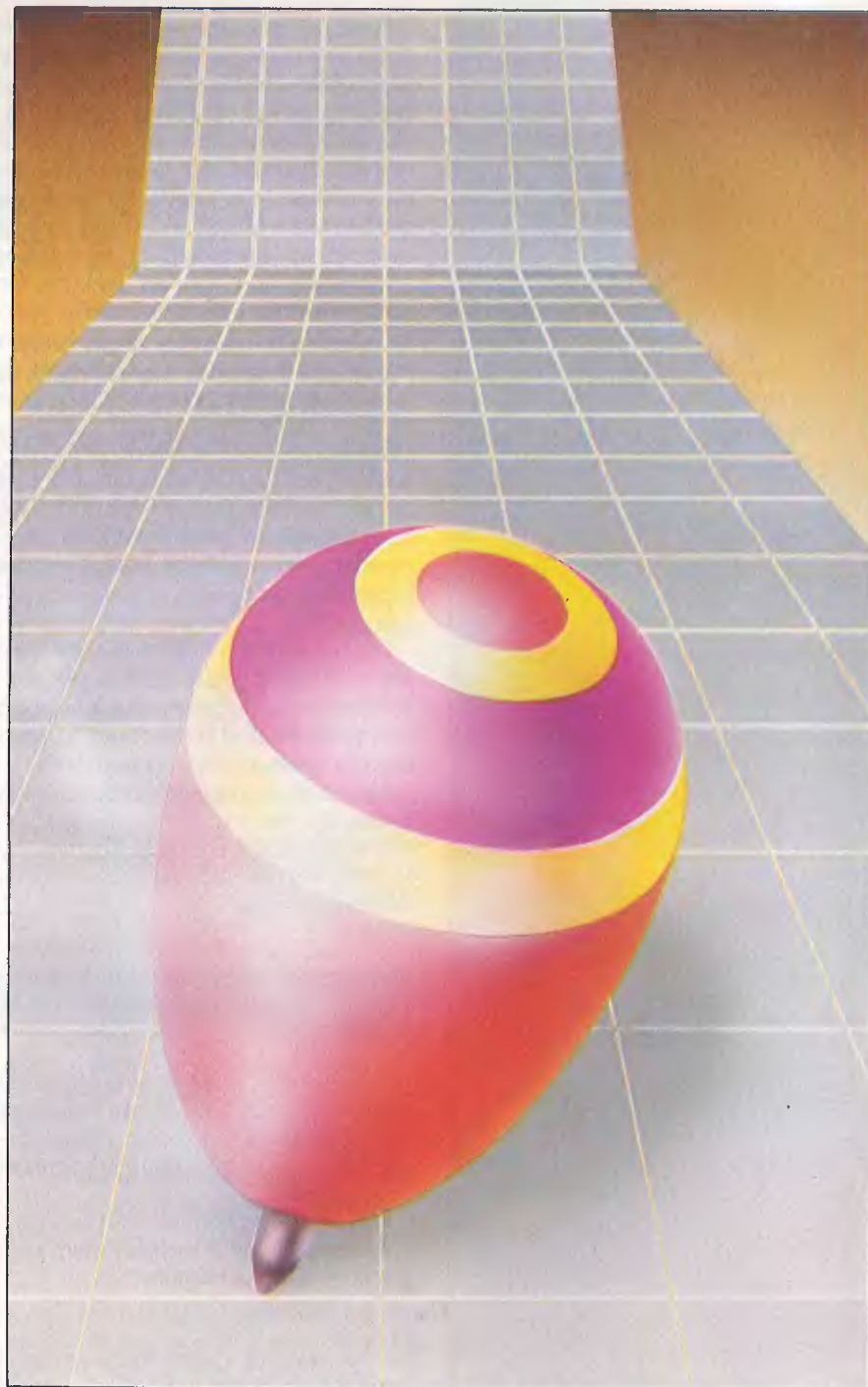
## Programa dos

Es un bucle FOR... NEXT similar al que utilizamos para hacer que nuestro nombre fuera creciendo. Pero ahora la variable de control «índice» irá decrementando desde la longitud de la palabra que hayamos introducido desde el teclado —en la línea 30— hasta que valga 1 y nos genere una sobcadena de una sola letra.

Al ejecutarlo parece que, comenzando por escribir la palabra completa la primera vez que se realiza el bucle, vamos quitando la primera letra en cada vuelta hasta quedarnos con una solamente.

En esta ocasión empleamos la función RIGHT\$, línea 70, y por eso es por lo que la

```
10 REM PROGRAMA II
20 CLS
30 INPUT "DIME UNA PALABRA: "; pala
bra$
40 PRINT
50 longitud=LEN(palabra$)
60 FOR indice=longitud TO 1 STEP -1
70 fragmento$=RIGHT$(palabra$,indice
$)
80 PRINT fragmento$
90 NEXT indice
100 PRINT
```



letra que desaparece es la del lado contrario: comenzamos a contar y seleccionar letras desde la primera de la derecha, tal como hemos dicho.

Imaginamos que si ha comprendido el modo de trabajo del programa anterior no tendrá ninguna pega en hacer lo mismo con éste, ya que salvo la nueva función todo lo demás es muy parecido.

Sólo nos queda decir sobre RIGHT\$ y LEFT\$ que su segundo parámetro sólo admite valores comprendidos entre 0 y 255. Y esto, a parte del funcionamiento interno del ordenador (no nos vamos a meter en él), tiene su lógica. ¿Alguna serie de caracteres tiene longitud negativa o compuesta por menos de «cero» letras? Es evidente que no, ¿verdad? El literal

más pequeño es la cadena vacía (""), y se escribe:

```
PRINT LEN("")
```

obtendrá como resultado que está compuesta por «ninguna» letra, o lo que es lo mismo que su longitud es cero.

Además, el **Amstrad** sólo admite expresiones literales que tengan como máximo 255 caracteres (una limitación). Así que no es ninguna novedad decir que una subcadena no puede tener más de 255 símbolos.

Pero de todas formas, compruebe qué pasa si llegamos a una línea que contenga una de estas funciones cuyo segundo parámetro no está comprendido entre estos valores. Te cleee:



fragmento\$=LEFT\$("PEPE",-32)

o:

fragmento\$=RIGHT\$("PEPE",328)

En ninguno de los dos casos engañamos a nuestro ordenador (además tampoco es esa nuestra intención). Nos acaban de aparecer en la pantalla sendos mensajes de error:

Improper argument

nos está indicando que estamos pasando como parámetros de una función unos valores que no están de acuerdo con la propia definición de la misma.

Además de estas dos, existen otras muchas que nos van a permitir manipular cadenas. El **Amstrad** es uno de los ordenadores que tiene mayor potencia y versatilidad en este campo.

Puede ocurrir que en algún momento necesitemos seleccionar una o varias letras del contenido de una cadena pero no a partir de la derecha o la izquierda, sino de una posición cualquiera dentro de la misma.

Por ejemplo, suponga que queremos escribir una palabra, pero en vez de horizontalmente necesitamos que aparezca en sentido vertical, como si fuera escritura china.

Si conocemos su contenido, o valor, no hay problema ya que podemos deletrearla nosotros mismos y luego imprimirla letra a letra, como en el programa 3.

### Programa tres

Con él conseguimos que aparezca en la pantalla la palabra «GATOS» en la cuarta columna y de arriba a bajo. Hemos dividido «GATOS» en cada uno de sus caracteres y los hemos ido escribiendo uno a uno y en filas diferentes.

```
10 REM PROGRAMA III
20 CLS
30 PRINT TAB(4) "G"
40 PRINT TAB(4) "A"
50 PRINT TAB(4) "T"
60 PRINT TAB(4) "O"
70 PRINT TAB(4) "S"
```

Las colocamos en la cuarta columna mediante el atributo TAB(4) que acompaña a cada una de las sentencias PRINT que dan forma al programa.

Al final obtendremos el resultado deseado, pero nos parece que el método empleado no es el más eficaz y por supuesto es bastante poco elegante: es rudimentario, en pocas palabras.

Sin ir más lejos, cuando no se trate de un valor alfanumérico conocido sino que sea el contenido de una variable de este tipo, ya nos empiezan a surgir problemas. No conocemos ni su composición ni su longitud ni nada de nada sobre él. Entonces, ¿cómo hacerlo?

El Basic de **Amstrad** es uno de los más poderosos en cuanto a tratamiento y manejo de cadenas. Además, siempre responde a tan favorable propaganda. Y en este caso también.

Se pone a nuestra disposición una instrucción que nos permitirá hacer fácil este problema que ahora nos aparece como insalvable:

MID\$(nombre\$,posición,caracteres)

función que nos dará una cadena, formada por el número de caracteres indicado por «caracteres» cogidos a partir del que está colocado en la posición «posición» dentro de lo que contenga en este momento la variable alfanumérica «nombre\$».

Si teclea:

PRINT MID\$("CASA",3,1)

aparecerá ante nuestros ojos un carácter (último parámetro) tomando a partir del tercero de la expresión literal «CASA». Resumiendo, veremos una «S».

Cuando escriba:

PRINT MID\$("FRUTA",2,3)

la función nos devolverá una cadena formada por tres letras tomadas a partir de la segunda de la palabra «FRUTA». O sea, «RUT».

Y así sucesivamente. ¿Intuye la manera de aplicarla a nuestro anterior problema? Ahora descompondremos la variable literal en todas y cada una de las letras, o símbolos, con toda facilidad utilizando esta nueva herramienta. Bastará con ir recorriendo el valor que contenga mediante un bucle FOR... NEXT, por ejemplo, cuya variable de control comenzará por 1 y llegará hasta la longitud de la palabra introducida para escribir en vertical.

El programa 4 será un buen sustituto del anterior. Observe las diferencias.

### Programa cuatro

Vea que en él la encargada de deletrear la palabra que hemos introducido al ordenador mediante el INPUT de la línea 30, es la función MID\$ de la 70. No pase por alto la manera de utilizarla.

Con el bucle, estamos seleccionando cada una de las letras de la expresión alfanumérica «palabra\$» de una en una, según nos indica el tercer parámetro de MID\$, comenzando por la inicial, ya que en la primera pa-

```
10 REM PROGRAMA IV
20 CLS
30 INPUT "ESCRIBE UNA PALABRA: ",palabra$
40 PRINT
50 longitud=LEN(palabra$)
60 FOR posicion=1 TO longitud
70 letra=MID$(palabra$,posicion,1)
80 PRINT TAB(20)letra$
90 NEXT posicion
100 PRINT
```

# Primeros PASOS

sada la variable de control vale 1, y terminando por la que ocupa el lugar dado por la «longitud» de la palabra tecleada, o sea, la última.

Así no tendremos problemas ni por el contenido ni por la longitud ya que siempre vamos a poder determinarlos inequívocamente.

Le sugerimos que intente variar el programa para que la palabra se escriba de abajo para arriba o de cualquier otra forma que se le ocurra.

Para terminar le dejamos con el programa 5. Es un ejemplo muy sencillo pero en el que vamos a combinar varias funciones de tratamiento de cadenas.

### Programa cinco

Su misión consiste en tomar una frase que nosotros le vamos a dar. Analizarla, manipularla y respondernos con el número de letras y palabras de que consta.

Le aseguro que si nos ha seguido detenidamente, no encontrará ninguna dificultad cuando se ponga a ver qué es exactamente lo que hace y cómo lo hace. ¡No incluimos nada nuevo!

```
10 REM PROGRAMA V
20 CLS
30 INPUT "ESCRIBE CUALQUIER FRASE. ",frase$
40 PRINT
50 longitud=LEN(frase$)
60 IF longitud=0 THEN letras=0:palabras=0:GOTO 120
70 FOR posicion=1 TO longitud
80 letra=MID$(frase$,posicion,1)
90 IF letra=CHR$(32) THEN letras=letras+1:espacio=0
100 IF (letra=CHR$(32) OR posicion=longitud) AND espacio=0 THEN palabras=palabras+1:espacio=1
110 NEXT posicion
120 CLS
130 PRINT "LA FRASE ";frase$
140 PRINT
150 PRINT TAB(10) " CONTIENE: "
160 PRINT
170 PRINT TAB(10) palabras; " PALABRAS "
180 PRINT
190 PRINT TAB(10) letras; " LETRAS "
200 PRINT
```

Pero no nos quedaremos aquí. Aunque ya estemos en verano y nos apelezca un «**mon-tón**» refrescarnos en la piscina, todo es compatible. Continuaremos descubriendo nuevas herramientas que nos hagan estar entre las «**cadenas**» no encadenados, sino como «**peces en el agua**». Refrescantes saludos.



# AMSTRAD CPC - 464

# AMSTRAD



## ORDENADOR

### SERIE CPC

#### UNIDAD CENTRAL. MEMORIAS

- Microprocesador Z80A - 64K RAM ampliables - 32K ROM ampliables
- **TECLADO** • Teclado profesional con 74 teclas en 3 bloques - Hasta 32 teclas programables - Teclado redefinible
- **PANTALLA** • Monitor RGB verde (12") o color (14")

	Normal	Alta Res.	Multicolor
Col x líneas	40 x 25	30 x 25	20 x 25
Colores	1 de 27	2 de 27	16 de 37
Puntos	320 x 200	640 x 200	160 x 2

- Se pueden definir hasta 8 ventanas de texto y 1 de gráficos • **SONIDO**
- 3 canales de 8 octavas moduladas independientemente - Altavoz interno regulable - Salida estéreo • **BASIC**
- Locomove BASIC ampliado en ROM Incluye los comandos AFTER y EVERY para control de interrupciones

### AMSTRAD CPC 464

- **CASSETTE** • Cassette incorporada con velocidad de grabación (1 ó 2 Kbaudios) controlada desde Basic • **CONECTORES**
- Bus PCB multiuso, Unidad de Disco exterior, paralelo Centronics, salida estéreo, joystick, lápiz óptico, etc.
- **SUMINISTRO** • Ordenador con monitor verde o color - 8 cassettes con programas - Libro "Guía de Referencia BASIC para el programador" - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA

**TODO POR** 59.900 Pts. (monitor verde)  
90.900 Pts. (monitor color)

### AMSTRAD CPC 6128

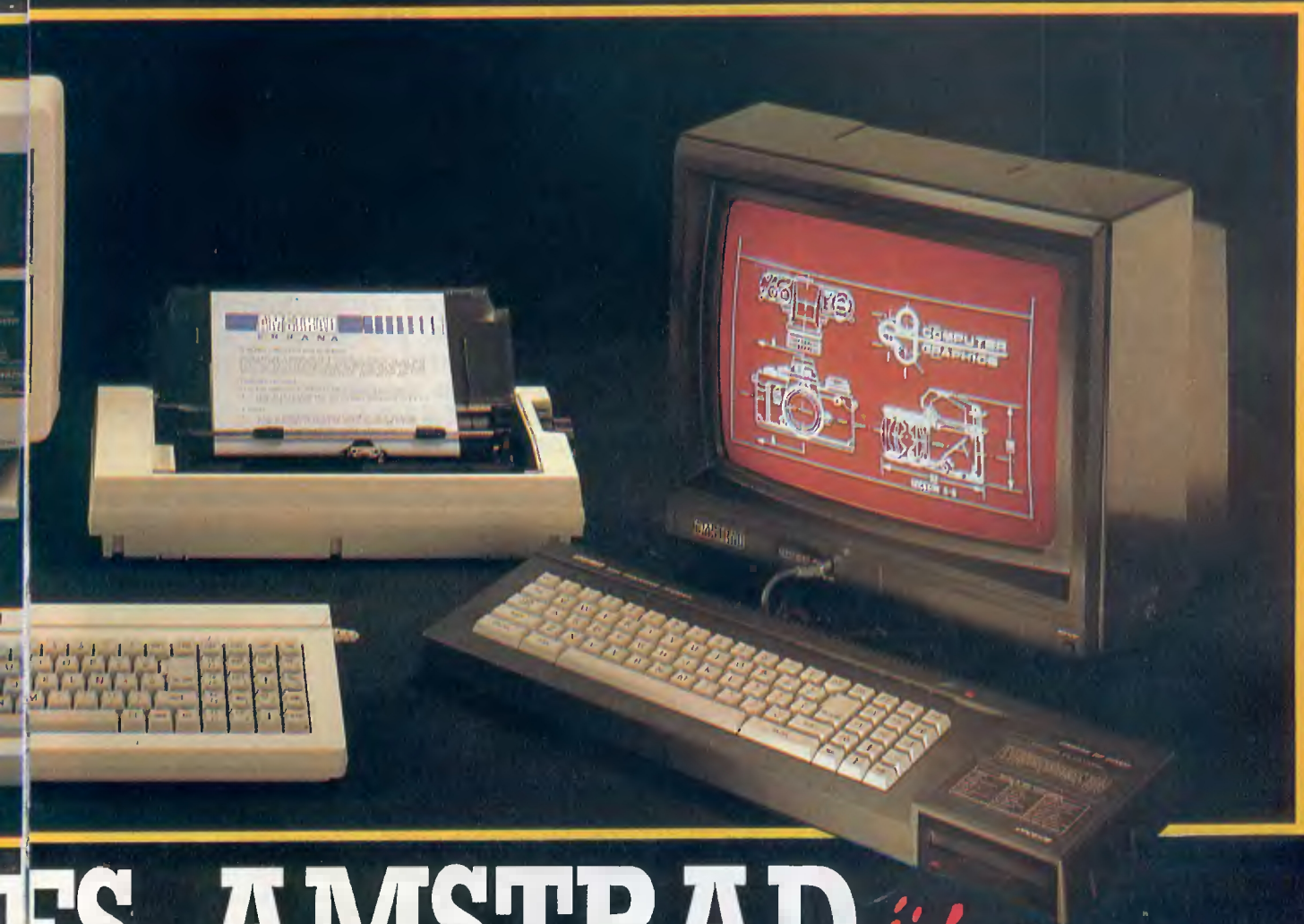
- **UNIDAD DE DISCO** • Unidad incorporada para disco de 3" con 180K por cara • **SISTEMAS OPERATIVOS**
- AMSDOS, CP/M 2.2, CP/M Plus (3.0)
- **CONECTORES** • Bus PCB multiuso, paralelo Centronics, cassette exterior, 2ª Unidad de Disco, salida estéreo, joysticks, lápiz óptico, etc.
- **SUMINISTRO** • Ordenador con monitor verde o color - Disco con CP/M 2.2 y lenguaje DR LOGO - Disco con CP/M Plus y utilidades - Disco con 6 programas de obsequio - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA.

**TODO POR** 84.900 Pts. (monitor verde)  
119.900 Pts. (monitor color)



# PCW - 8256

# AMSTRAD CPC - 6128



# ES AMSTRAD

## AMSTRAD PCW 8256

### UNIDAD CENTRAL. MEMORIAS

• Microprocesador Z80A - 256K RAM de las que 112K se utilizan como disco RAM

• **TECLADO** • Teclado profesional en castellano (ñ, acento...) de 82 teclas

• **PANTALLA** • Monitor verde de alta resolución - 90 columnas X 32 líneas de texto

• **UNIDAD DE DISCO** • Disco de 3" y 173K por cara - Opcionalmente, 2.ª Unidad de Disco de 1 Mbyte integrable

• **SISTEMA OPERATIVO** • CP/M Plus de Digital Research • **IMPRESORA** •

Alta calidad (NLQ) a 20 c.p.s. - Calidad estándar a 90 c.p.s. - Papel continuo u hojas sueltas - Alineación automática del papel - Caracteres normales, comprimidos, expandidos, control del paso de letra (normal, cursiva, negrita, subíndices, superíndices, subrayado, etc)

• **OPCIONES** • Kit de Ampliación a 512K RAM y 2.ª Unidad de Disco Interface Serie RS 232C y paralelo

Centronics • **SUMINISTRO** • Ordenador completo con teclado, pantalla, Unidad de Disco e Impresora - Discos con el procesador de Texto LocoScript, CP/M Plus, Mallard, BASIC, DR. LOGO y diversas utilidades - Manuales en castellano - Garantía Oficial AMSTRAD ESPAÑA.

**TODO POR 129.900 Pts.**



Los más prestigiosos paquetes de Software Profesional, en formato AMSTRAD... a "precios AMSTRAD"

## ¡Increíble!!

Existe también la versión **PCW 8512** con **512K RAM** y la 2.ª Unidad de Disco de 1 Mbyte incorporada. **PVP. 169.900 Pts.**

\* El **PCW 8256** puede utilizarse como terminal y en comunicaciones.

El I.V.A. no está incluido en los precios.

**NOTA:** Es muy importante verificar la garantía del aparato ya que sólo **AMSTRAD ESPAÑA** puede garantizarle la ordenada reparación y sobre todo materiales de repuesto oficiales (Monitor, ordenador, cassette o unidades de discos).

## AMSTRAD

ESPAÑA

Avda. del Mediterráneo, 9. Tels. 433 45 48 - 433 48 76.  
28007 MADRID

Delegación Cataluña: Tarragona, 110 - Tel. 325 10 58.  
08015 BARCELONA



# PROGRAMACION ESTRUCTURADA

*Respecto a la programación, yo creo que hay una vieja pregunta que está permanentemente en el aire: ¿es un arte o una ciencia? Los programadores profesionales, que crean maravillas como «Knigh Lore», o como «Nucleus», ¿son científicos que siguen rígidamente un disciplinado y estudiadísimo método de trabajo? o, por el contrario, ¿son iluminados que, a golpes de imaginación e intuición, construyen obras geniales que elevan a un estúpido puñado de circuitos a la categoría de herramienta indispensable del siglo XX? Dos preguntas, y una sola respuesta: Programación estructurada.*







icen las malas lenguas que la programación y el ma-soquismo son parientes muy, pero que muy cercanos.

Los profanos en el esotérico (dicen ellos) arte de la programación, tienen en la mente la imagen de una persona encadenada a un terminal, con el pelo revuelto y mirada vesánica, envuelto en las hojas de un interminable listado y al cual casi se le puede oír funcionar el cerebro, tratando sin éxito de resolver abstrusos problemas completamente alejados del común de los mortales.

Bien, pues es cierto. En general, los programadores son así: están a caballo entre la cadena de producción y montaje y la más refinada expresión del arte.

Al leer las palabras producción y montaje, supongo que se habrá encendido inmediatamente la luz de alerta roja: «Hum..., producción, esto huele a dinero».

¡Bingo! A *mucho* dinero; el volumen de negocios, para entendernos, de millones, que una casa de software de las grandes, como **Microsoft**, mueve, es espeluznante.

No es muy sorprendente que gran cantidad de cerebros lúcidos hayan dedicado mucho tiempo a averiguar la forma de maximizar la producción de los programadores, y, a la vez, a minimizar los costos.

## EL NACIMIENTO DE UN METODO

Estos «pensadores» descubrieron que la parte creativa de la programación mejor no tocarla, pues es difícilmente mensurable, pero la parte de producción, eso es otro cantar.

Hoy en día, el hardware está «tirado» de precio, y cada vez más. Sin embargo, los programas cuestan, *vox populi*, «una pasta» y, mira por donde, de todo este dineral, la mayor parte no se va en diseñar y escribir un programa, sino en corregir los errores que tenga, y en mantenerlo actualizado, para que responda a las cambiantes necesidades de un mercado muy inestable, sujeto a modas, a la vez que posea la generalidad suficiente para poder adaptarlo, personalizándolo, a los gustos de cada cliente.

Como a los programadores, sobre todo a los asalariados, todo este edificio de sutilezas les importa un bledo, no hubo más remedio que dotarles de una disciplina profundamente estudiada de optimización de costos, basada, parte en la planificación, y parte en la experiencia extraída de la labor diaria de los mismos programadores.

Es el momento de introducir las dos palabras que dan título a este artículo: Programación Estructurada.

La frase, bastante cursi por cierto, hace referencia directa a otro par de cositas: **Niklaus Wirth** y **Pascal**; un hombre y su lenguaje.

En los últimos estertores de los 60, **Wirth** dijo muy alto lo que casi todo el mundo pensaba y admitía en el desván de la programación: «Señores, no tenemos idea de cómo programar, y los lenguajes más en uso hoy, Basic y Fortran, son un churro.»

La verdad es que lo dijo con bastante más «mano izquierda» que el autor de este artículo, y, de tal forma, que la caterva de escépticos que inmediatamente alzó el vuelo ante él, pudieron tocar, como Santo Tomás, algo tangible que resolvía el asunto de forma concluyente: el Pascal.

## LLEGA PASCAL

Nosotros, en mi opinión por desgracia, hemos mamado de los pechos del Basic desde que nacimos a la informática, y así nos va; sin embargo, aunque Pascal sea un asunto aparte, podemos aplicar su lógica y sus postulados de base, de diseño, a los programas que escribamos en Basic, salvando las distancias.

Wirth basó el Pascal en una serie de principios muy simples, y muy lógicos, teniendo en cuenta la altura a la que se encontraba el hardware en su época (hace muchos, muchos años):

- No usar el lenguaje máquina más que cuando sea absolutamente necesario. Vivan los lenguajes de alto nivel.

- Los únicos lenguajes viables para programación seria, por razones de eficiencia (velocidad), son los compilados.

- El compilador debe facilitar al máximo la labor del programador: una forma de hacerlo es crear un lenguaje extremadamente rígido, no dejando al azar ningún tipo de ambigüedad: o es, o no es.

- Partiendo de la base de que el programador tiende a ser desorganizado y variable, el lenguaje debe imponer una monolítica disciplina a la hora de crear un programa.

- La fase de codificación del programa debe ser la última en realizarse: el diseño debe materializarse y crecer, pues, sobre papel.

- Es más fácil resolver un gran problema dividiéndolo en pequeñas partes, y éstas a su vez en otras, hasta alcanzar el caso trivial, que abordar el asunto, confusamente, como un todo.

El problema con el Basic es que no todos los postulados anteriores le son aplicables: Basic no es compilado, por lo menos el del **Amstrad**, ni es rígido: permite al programador todo tipo de libertades, favoreciendo la creación de programas confusos e indescifrables.

## ESTRUCTURAR EN BASIC

No obstante, algo podremos hacer. Vamos a partir de la base de que, en nuestro caso, el programador *sí* sabe lo que está haciendo, aunque esto implica, créanme, que ya hayamos empleado unas cuantas horas en el diseño y comprensión de la lógica del problema.

Imaginemos que se necesita una agenda, y que no estamos nada dispuestos a comprar una. Sólo queda una alternativa: hacérnosla.

Primer problema: ¿cuál va a ser el tamaño de los datos que tengamos que manejar?

En apariencia, la pregunta es un tanto sorprendente, pero mírelo de esta manera: si vamos a meter el equivalente en datos de una guía telefónica, para almacenarlos necesitamos docenas de discos o un disco duro, o sea, apaga y vámonos, mientras que si el volumen de datos es razonable, podemos conformarnos con almacenarlos en un disco normal, en humildes ficheros secuenciales. También el asunto del tamaño afecta al algoritmo de búsqueda y ordenación que tomemos, como luego se verá.

Así, no hemos escrito una línea de programa, pero ya nos hemos encontrado con el problema fundamental del programador: ¿qué estructura de datos es la más conveniente?

Al decidir que trataremos con cantidades «razonables», parece lógico usar matrices alfanuméricas para al-



macenarlos en la memoria, así conseguiremos una mayor velocidad y eficiencia en las previsibles operaciones de ordenación y búsqueda.

Un enfoque más general y riguroso, que permitiera tratar con cualquier volumen de datos, no nos compensaría el tiempo invertido en desarrollarlo (ni el dinero) para este caso concreto.

Una vez decidido el tipo de datos, matrices alfanuméricas, y dónde guardarlos cuando hayamos terminado con ellos, ficheros de disco secuenciales, podemos plantearnos qué, y cómo, va a funcionar la agenda. Insisto: ni una línea de programa hasta ahora.

### **POR EL CAOS HACIA EL ORDEN**

Veamos, ¿qué debe hacer la agenda? Sí, ya sé, ni idea. Bueno, a estas alturas, yo tampoco. La tabla de salvación de angustiosas situaciones como la presente, una de ellas, es escribir todas las ideas que se nos ocurran en cualquier orden; luego, ya se le dará una estructura y una jerarquía.

Un, dos, tres, probando:

1. Tipos de datos: matrices o matriz para operar con ellas en memoria. Mayor velocidad. Guardar en ficheros secuenciales de disco, tipo texto.

2. Lectura de los datos del disco. Introducirlos en matriz o matrices correspondientes.

3. Aceptar nuevos datos. Comprobar su validez e introducirlos en la matriz correspondiente.

4. Fin de la sesión. Todos los datos al disco.

5. Inicialización de variables y ficheros.

6. Posible presentación en forma de menú de opciones.

7. Impresión de los datos por pantalla y/o impresora.

8. Ordenación de los datos por el método más rápido posible, en función del número de datos que tengamos que tratar.

Leyendo atentamente este «caos», se puede dar un paso más en el diseño de la agenda, agrupando todo esto en categorías:

a) Tipos de datos. 1

b) Inicialización. 5

c) Presentación. 6

d) Proceso de los datos. 2, 8, 4



e) Entradas/salidas. 3, 7

Tenemos 5 apartados, de los cuales el «a)» puede englobarse dentro del «b)» a la hora de escribir el programa, ya que está ahí puesto, más que nada, para nuestra información.

Estamos listos para dar otro paso adelante: escribir el cuerpo principal del programa, encargado de manejar una serie de subrutinas que van a realizar las tareas anteriores. Sería, más o menos, así:

REM CUERPO PRINCIPAL DEL  
PROGRAMA AGENDA

GOSUB inicializar  
GOSUB presentación  
GOSUB menú  
GOSUB aceptadatos  
GOSUB metedatos  
GOSUB ordenadatos  
GOSUB imprimedatos  
GOSUB finprograma  
END

Bueno, algo más claro sí que está; sin embargo, las categorías no están bien escogidas: algunas subrutinas dependen de la elección que se ha-

ga en el menú de opciones; vamos a estructurar de nuevo:

REM CUERPO PRINCIPAL DEL PRO-  
GRAMA AGENDA

GOSUB inicializar  
GOSUB presentación  
GOSUB MENU

ON elección GOSUB aceptadatos,  
metedatos, ordenadatos, imprime-  
datos, finprograma

END

El programa va tomando forma, pero todavía adolece de un defecto esencial y bastante común: todas las subrutinas, después de menú, sólo se ejecutarán una vez. Hay que instruir al programa para que se detenga cuando finalicemos. Para ello, la rutina «finprograma» debe interactuar con el programa principal, mediante un flag que causará la salida de un bucle WHILE...WEND, de esta manera:

REM CUERPO PRINCIPAL DEL PRO-  
GRAMA AGENDA



GOSUB inicializar  
WHILE NOT fin  
GOSUB presentación  
GOSUB MENU

ON elección GOSUB aceptadatos,  
metedatos, ordenadatos, imprime-  
datos, finprograma

WEND  
END

Yo, personalmente, recuerdo mis días de «programador spaguetti», cuando me sentaba al teclado y, lleno de un ardor casi religioso, codificaba como un loco programas de este tipo, intentando hacerlo todo de una vez. Ni que decir tiene que el producto final era, aparte de un maremagnum de indiscriminados «GOTOS», una verdadera obra de arte de criptografía, que no de la programación, envidia eterna de los más pulcros y hábiles artífices de jeroglíficos del Antiguo Egipto.

Como uno, por muy programador que sea, también tiene su corazoncito, no puedo por menos que gritar de júbilo al ver que está perfectamente clara la estructura y funcionamiento de un programa sin ni siquiera haber tocado el ordenador.

### VENTAJAS DE LA PROGRAMACION ESTRUCTURADA

Espero que los lectores, amablemente, perdonen este explosivo cántico a las estructuras, tema no muy romántico, pero de indudable utilidad si uno se fija en un punto esencial: nuestro programa, va a **FUNCIONAR SEGURO**, sin habernos tomado la molestia de definir una sola rutina, y, además, está pensado de tal forma que permite, a su vez, dos cosas:

1. Las rutinas servirán para cualquier programa que cumpla una mínima semejanza con nuestra agenda: que maneje matrices alfanuméricas y ficheros secuenciales. En efecto, una rutina «ordenadatos» funcionará haya lo que haya en la matriz a ordenar.

2. Las rutinas son intercambiables en función de las necesidades del programa.

Por ejemplo, para ordenar 100 datos, nos basta un algoritmo simple del tipo burbuja; para ordenar 1000,

tendríamos que recurrir a algo más eficiente, como el «quicksort» (en castellano, «ordenación rápida»), **PERO SOLO HABRA QUE CAMBIAR ESA RUTINA; EL RESTO DEL PROGRAMA PERMANECE INVARIABLE.**

Casi sin darnos cuenta, nos hemos introducido en otro asunto clave de la programación estructurada, de la que hemos hablado por encima en otras ocasiones: la MODULARIDAD, cuyo estudio en profundidad, junto a la codificación definitiva de las **subrutinas** del program agenda, dejamos para la segunda parte de este artículo.

Antes de despedirme, y asumiendo un papel paternalista en el mejor sentido de la palabra, si es que lo tiene, creo que sería una buena idea que el lector intentara, siguiendo el método expuesto en este artículo, codificar y estructurar el programa, con el doble objeto de practicar en algo clave para cualquier programador, y poder comparar su solución con la que propondremos en el próximo artículo; puede que la suya sea más elegante y eficiente que la nuestra. Siempre hay más de una forma de hacer las cosas.

## ¿Te falta algún número?

**S**i estás interesado en algún número de los ya publicados por Microhobby Amstrad, realiza hoy mismo tu pedido porque ya hay algunos ejemplares agotados.

No pierdas la oportunidad de disponer de la mejor obra publicada sobre ordenadores Amstrad. En todos sus números encontrarás interesantes artículos de iniciación, pokes, trucos, curso de código máquina, etc...

¡No te pierdas detalle!

*Recorta o copia el cupón que aparece cosido en las páginas de la revista.*





# COMANDO

**Tras largos meses de espera, por fin llega el famoso comando a las pantallas de Amstrad. Empuja el joystick con decisión y dispara a todo lo que se mueva a tu alrededor.**



¿Qué fue primero, el huevo o la gallina? Este misterioso enigma, que ha traído de cabeza a la humanidad desde el Renacimiento, nunca ha admitido el menor género de duda en el mundo del software.

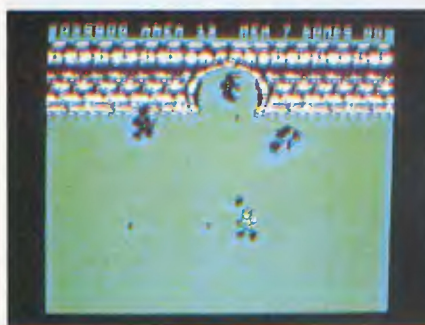
Siempre que una película alcanza altas cotas de éxito, y su fama se extiende por todos los países, las casas de software se apresuran a lanzar la versión de la misma, en juego de ordenador.

En nuestro largo camino por el mundo del software, hemos visto de todo; desde las adaptaciones hechas con precipitación sin cuidar demasiado el producto, hasta obras de un inestimable valor.

No obstante, la tónica general de este tipo de productos, es conseguir una rápida adaptación para aprovechar el periodo de éxito taquillero de la película, y de esta forma poder vender el máximo de copias basándose en el nombre del film.



Compatible: CPC/464, CPC/664 y CPC/6128



Como política comercial, la cosa no está nada mal, y seguro que las cifras de ventas son altas, y producen pingües beneficios.

Pero, en cambio, atendiendo a la calidad del software, los programas, debido principalmente a la premura con que se realizan, no son obras maestras ni mucho menos.

Con Comando, el orden de los acontecimientos se ha visto alterado profundamente. Por primera vez (y esperemos que no sea la última) es un juego de ordenador el que da pie a la creación de una película.

Si siempre había sido el cine el antecedente del software, sin embargo, Comando ha extendido su poder desde los monitores de ordenador, hasta las gigantescas pantallas de las salas de exhibición.

El principal responsable de todo este revuelo un juego de acción trepidante, en el que todo se desarrolla a base de fusil ametrallador y granada de mano.

El creador del mismo es la casa nipona CAPCOM, cuyo campo de acción son los juegos de máquinas de billares, consiguiendo un producto



de espectacular colorido y efectismo altamente realista.

Con Comando entramos en el mundo de los héroes de Hazañas Bélicas. En misión suicida un solo hombre se enfrenta a todo un ejército de enemigos armados hasta los dientes.

Con el más puro espíritu legionario, nuestro héroe continuará avanzando hasta llegar al polvo y destruirlo, o morir en el empeño.

Joe, soldado de élite experto en lucha de guerrillas, y misiones que requieren un alto grado de capacitación física, es soltado por un helicóptero en los límites de la zona controlada por la guerrilla.

Depositado en un claro de la espesa vegetación, el estruendo del aparato pone en guardia a las pa-







trullas de vigilancia que circundan la zona.

En un abrir y cerrar de ojos, empiezan a aparecer hasta de debajo de las piedras fuerzas de la guerrilla, dispuestas a liquidar al intruso que se interna en su zona.

Solamente las inmejorables condiciones de Joe, pueden hacer que supere estos primeros inconvenientes sin el menor problema.

Entrenado en las más modernas técnicas de lucha cuerpo a cuerpo, su rapidez de reflejos y excelente puntería hacen caer como moscas a sus primeros contendientes.

El uso continuo de su fusil ametrallador, limpia de estorbos las inmediaciones de su imparable marcha; unos metros más adelante, el conti-

nuo tableteo de una ametralladora denuncia un peligro más serio.

Tras un murete de sacos terreros, la pieza de tiro y sus servidores llenan de plomo el aire a nuestro alrededor. Es el momento en que Joe extrae una granada de mano de su cinturón, y, lanzándola certeramente, hace saltar por los aires la ametralladora y sus sirvientes.

El próximo desafío: cruzar el puente de la carreta general; sitio que, por la escasa vegetación y las patrullas motorizadas que circulan por ella, hacen extremadamente peligroso su paso.

De nuevo en la jungla, puestos de franco tirador nos acechan; movilidad y unas ráfagas de fusil ametrallador son suficientes para superarlos.

Tras ellos se erigen las puertas del primer fortín, las cuales se abren dando paso a una verdadera maraña de enemigos. Joe empuña con la izquierda el fusil ametrallador y sin dejar ni un momento de disparar, lanza granadas con la mano derecha.

Dentro de los muros del fortín, tanquetas y soldados atrincherados nos aguardan; tras ellos los barracones escupen por sus puertas compañías enteras.

Superados estos casamatas, bunkers y acuartelamientos subterráneos protegen las puertas de la zona de máxima seguridad.

Los próximos peligros que esperan a Joe sólo son para jugadores expertos. Cualquier mortal sucumbiría a las puertas del primer fortín, los superhombres pasarán.

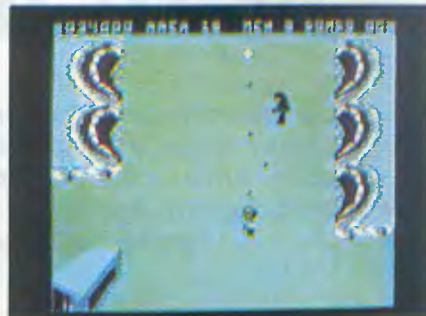
El juego de Elite, casa responsable de la versión para **Amstrad**, llega un poco tarde, casi cuando la película ha pasado a la historia, y con varios meses de retraso respecto a las otras versiones para ordenador.

El programa está realizado utilizando el modo de 16 colores. Los gráficos realizados con una técnica bastante «naïf», resultan infantiles, al compararlos con lo visto en versiones del mismo juego para otros ordenadores.

El ángulo de visión está enfocado como si de una vista aérea se tratara, quedando los gráficos de los soldados dotados de una extraña constitución física.

Al ver las figuras desde arriba, los soldados resultan patiocortos, produciendo un gracioso efecto al ponerse en movimiento; sólo se ve cabeza, espalda y pies.

## Mister JOYSTICK



El tamaño de los guerrilleros, de dimensiones reducidas (2,5 cm de altura), contribuye a reforzar más aún el efecto infantil de los gráficos: la animación le sigue de cerca.

Una incesante música, acompaña las andanzas de Joe, mientras éste se abre paso entre la maraña de enemigos que le acosan.

Un juego de constante acción, en el que hemos de permanecer constantemente en movimiento, disparando sin cesar a nuestro alrededor, manteniéndonos preparados en todo momento, para las sorpresas que nos esperan tras cada zona defensiva superada.





# ABRE LAS VENTANAS DE TU AMSTRAD

*Continuamos esta semana hablando de la impresión en pantalla. En este capítulo explicaremos cómo se pueden definir ventanas desde código máquina y cómo seleccionar el color para cada una de ellas.*



Como todos sabemos, cada una de las ocho **ventanas** que se pueden obtener en el **Amstrad**, lleva asociado un canal a través del cual puede reconocerse cada una de ellas en el momento deseado.

Además dichas ventanas pueden llenarse con colores distintos previamente definidos por nosotros, seleccionando el canal correspondiente.

Lo primero que se hace al entrar en la rutina, es seleccionar el modo de pantalla con el que se va a trabajar. En este caso elegiremos el modo cero para obtener mayor número de colores, ya que en este modo se pueden definir hasta 16 colores distintos.

Una vez seleccionado el modo de trabajo, colocaremos el color del borde de la pantalla con el color que se desee.

Todas las llamadas al firmware utilizadas hasta el momento, ya han sido estudiadas en anteriores capítulos.

El siguiente paso de programa, debe ser la definición de cada una de las **ventanas** que se deseen obtener en pantalla. Para ello debemos seleccionar en primer lugar el canal correspondiente a cada una de dichas ventanas.

La selección de canales la realizaremos mediante la siguiente llamada:

**SELECCIONA CANAL. #BBB4**

Selecciona el canal de impresión de la ventana correspondiente.

Condiciones de entrada.  
En el registro A se debe indicar el número de canal elegido.

Condiciones de salida.  
El registro A contiene el número de canal seleccionado previamente. El registro doble HL y todos los flags se corrompen. Los demás registros son preservados por dicha rutina.

Dado que el **Amstrad** posee 8 (0-7) **canales** de impresión en pantalla, dicha rutina se asegura de que el número de canal no sea mayor a 7, provocando una máscara con dicho número del acumulador.

Una vez elegido el canal correspondiente a la ventana, deberemos seleccionar la ventana propiamente dicha, para ello se efectuará una llamada al firmware que se encarga de producir dicho efecto. Esta llamada es la siguiente:

**SELECCIONA VENTANA. #BB66**

Selecciona una ventana. Los parámetros de entrada deben ser la primera y última columna y la primera y última fila de dicha ventana.

Condiciones de entrada.  
El registro H debe contener la primera columna, el registro D contendrá la última columna, el registro L debe contener la primera fila y el registro E la última.

Condiciones de salida.  
Se preservan todos los flags y registros excepto AF, BC, DE y HL.

## **El color del papel**

Seguidamente deberemos seleccionar el papel con el cual deseamos que se rellene dicha ventana.

Dado que ya se ha seleccionado el canal, todas las operaciones que se







hagan irán referidas a dicho canal hasta que no se especifique lo contrario.

Así pues, el papel que se selecciona en este momento será el que corresponda a la ventana anteriormente definida.

Sólo queda, pues provocar un «CLS» para que nuestra ventana se muestre con el color del papel elegido.

Estas dos últimas operaciones se realizan con llamadas al **firmware** que ya se han explicado en anteriores capítulos, por lo que no perderemos el tiempo en comentarlas de nuevo.

Para definir las restantes ventanas, se hace exactamente lo mismo que lo que se ha explicado anteriormente, únicamente debe variar el canal correspondiente a cada una de las ventanas y el color que se elija para ellas.

Como podemos comprobar en los listados Basic y Assembler del programa que estamos estudiando, existe una diferencia en los parámetros con los que se definen las ventanas. Esto es debido a que el sistema Basic utiliza como coordenadas iniciales, es decir como coordenadas de principio de pantalla, la (1,1).

## PROGRAMA BASIC

```

10 MODE 0
20 BORDER 9
30 WINDOW #1,1,10,1,5
40 PAPER #1,1
50 CLS #1
60 WINDOW #2,11,20,1,5
70 PAPER #2,2
80 CLS #2
90 WINDOW #3,1,10,20,25
100 PAPER #3,3
110 CLS #3
120 WINDOW #4,11,20,20,25
130 PAPER #4,4
140 CLS #4
150 WINDOW #5,1,5,6,19
160 PAPER #5,5
170 CLS #5
180 WINDOW #6,16,20,6,19
190 PAPER #6,6
200 CLS #6
210 PEN 7
220 LOCATE 8,9:PRINT "SIETE"
230 LOCATE 7,11:PRINT "VENTANAS"
240 LOCATE 9,13:PRINT "EN"
250 LOCATE 7,15:PRINT "PANTALLA"
260 WHILE INKEY$="" :WEND

```

Por el contrario el firmware utiliza como coordenadas de inicio de pantalla las (0,0), de ahí la diferencia existente entre los parámetros de los dos listados.

Quizá con un ejemplo práctico comprendamos mejor cuál es la diferencia existente. Si, por ejemplo, deseamos definir una ventana que ocupe la mitad superior de la pan-

## Código MAQUINA

talla trabajando en modo cero, en Basic deberíamos confeccionar el siguiente programa:

WINDOW #1,1,20,1,12

dicha ventana iría de la columna 1 hasta la columna 20 y de la fila 1 hasta la fila 12. Además el canal reconocido para la misma sería el canal 1.

Para ejecutar el mismo programa desde código máquina, deberíamos definir como condiciones iniciales las que se dan a continuación:

```

LD H,0 (Columna 1 en Basic)
LD L,0 (Fila en Basic)
LD D,19 (Columna 20 en Basic)
LD E,11 (Fila 12 en Basic)
CALL #BB66

```

Por supuesto, antes de haber definido dicha ventana deberíamos haber seleccionado el canal correspondiente, de lo contrario se tomaría el canal por defecto, que sería el que se hubiese elegido anteriormente.

Así pues, si dicha ventana se quisiera identificar con el canal 5, deberíamos haber escrito lo siguiente:

```

LD A,5 (Número de canal)
CALL #BBB4

```

## Elección de la pluma

Una vez definidas cada una de las ventanas, deberemos preparar la pluma con la cual deseamos escribir el texto en pantalla, para ello cargaremos en el acumulador el número de pluma y se llamará a la rutina del firmware que se encargue de realizar dicho trabajo.

Ahora bien, si no se le indica por qué canal deseamos imprimir dicho texto, lo hará por el canal seleccionado anteriormente.

Dado que el texto se debe imprimir en la ventana inicial de pantalla, es decir, en el trozo de pantalla en el cual no se ha definido ninguna de las anteriores ventanas, deberemos seleccionar el canal 0, que es el que se da a la ventana inicial cuando se inicializa el ordenador.



Así pues, deberemos cargar el acumulador con el valor cero y llamar a la rutina encargada de seleccionar la ventana.

Una vez hecho esto, estamos en condiciones de imprimir el texto deseado. Para ello cargamos en el registro doble HL las coordenadas en las cuales se debe imprimir el texto y en DE la dirección de dicho texto en la memoria.

Con estos dos registros dobles asignados, llamaremos a las rutina

PRINT que es la que se encargará de la impresión propiamente dicha.

Esta rutina ya se ha utilizado en anteriores capítulos y su funcionamiento es bien sencillo, ya que lo único que hace es imprimir caracteres a partir de la dirección de memoria indicada en HL, hasta encontrar el valor 255, lo cual hace que retorne al programa principal.

Además sólo entrar en dicha rutina se llama a la rutina del firmware que se encarga de colocar el cursor

en la posición indicada por el registro doble HL.

Así pues, una vez imprimido el primer texto, se asignarán nuevos valores de posición de pantalla y dirección de texto en memoria, y se llamará de nuevo a dicha rutina.

Una vez finalizada la impresión de los textos en pantalla, se llama a la rutina que espera a que se pulse una tecla, y cuando ésta se haya pulsado, retornamos al Basic.

Por último haremos una breve re-

## LISTADO DESENSAMBLADO

A000	10	ORG #A000			
	20				
	30	;MODE 0			
	40				
A000 AF	50	XOR A			
A001 CD0EBC	60	CALL #BC0E			
	70				
	80	;BORDER 9			
	90				
A004 010909	100	LD BC,#0909			
A007 CD38BC	110	CALL #BC38			
	120				
	130	;WINDOW #1,1,10,1,5			
	140				
A00A 3E01	150	LD A,1			
A00C CDB4BB	160	CALL #BBB4			
A00F 210000	170	LD HL,#0000			
A012 110409	180	LD DE,#0904			
A015 CD66BB	190	CALL #BB66			
	200				
	210	;PAPER #1,1			
	220				
A018 3E01	230	LD A,1			
A01A CD96BB	240	CALL #BB96			
	250				
	260	;CLS #1			
	270				
A01D CD6CBB	280	CALL #BB6C			
	290				
	300	;A CONTINUACION DEFINIREMOS			
	310	;LAS RESTANTES VENTANAS DE LA			
	320	;MISMA FORMA QUE HEMOS HECHO			
	330	;CON LA PRIMERA			
	340				
A020 3E02	350	LD A,2			
A022 CDB4BB	360	CALL #BBB4			
A025 21000A	370	LD HL,#0A00			
A028 110413	380	LD DE,#1304			
A02B CD66BB	390	CALL #BB66			
A02E 3E02	400	LD A,2			
A030 CD96BB	410	CALL #BB96			
A033 CD6CBB	420	CALL #BB6C			
A036 3E03	430	LD A,3			
A038 CDB4BB	440	CALL #BBB4			
A03B 211300	450	LD HL,#0013			
A03E 111809	460	LD DE,#0918			
A041 CD66BB	470	CALL #BB66			
A044 3E03	480	LD A,3			
A046 CD96BB	490	CALL #BB96			
A049 CD6CBB	500	CALL #BB6C			
A04C 3E04	510	LD A,4			
A04E CDB4BB	520	CALL #BBB4			
A051 21130A	530	LD HL,#0A13			
A054 111913	540	LD DE,#1319			
A057 CD66BB	550	CALL #BB66			
A05A 3E04	560	LD A,4			
A05C CD96BB	570	CALL #BB96			
A05F CD6CBB	580	CALL #BB6C			
A062 3E05	590	LD A,5			
A064 CDB4BB	600	CALL #BBB4			
A067 210500	610	LD HL,#0005			
A06A 111204	620	LD DE,#0412			
A06D CD66BB	630	CALL #BB66			
A070 3E05	640	LD A,5			
A072 CD96BB	650	CALL #BB96			
A075 CD6CBB	660	CALL #BB6C			
A078 3E06	670	LD A,6			
A07A 21050F	680	LD HL,#0F05			
A07D 111213	690	LD DE,#1312			
A080 CD66BB	700	CALL #BB66			
A083 3E06	710	LD A,6			
A085 CD96BB	720	CALL #BB96			
A088 CD6CBB	730	CALL #BB6C			
	740				
	750	;SELECCION DEL CANAL CERO			
	760	;PARA IMPRIMIR EL TEXTO			
	770				
A08B AF	780	XOR A			
A08C CDB4BB	790	CALL #BBB4			
	800				
	810	;PEN 7			
	820				
A08F 3E07	830	LD A,7			
A091 CD96BB	840	CALL #BB96			
	850				
	860	;IMPRESION DEL PRIMER TEXTO			
	870				
A094 210908	880	LD HL,#0809			
A097 11C9A0	890	LD DE,TXT1			
A09A CDBCA0	900	CALL PRINT			
	910				
	920	;LA IMPRESION DE LOS DEMAS			
	930	;TEXTOS ES IDENTICA A LA DEL			
	940	;PRIMERO			
	950				
A09D 210B07	960	LD HL,#070B			
A0A0 11CFA0	970	LD DE,TXT2			
A0A3 CDBCA0	980	CALL PRINT			
A0A6 210D09	990	LD HL,#090D			
A0A9 11D8A0	1000	LD DE,TXT3			
A0AC CDBCA0	1010	CALL PRINT			
A0AF 210F07	1020	LD HL,#070F			
A0B2 11D8A0	1030	LD DE,TXT4			
A0B5 CDBCA0	1040	CALL PRINT			
	1050				
	1060	;ESPERA LA PULSACION DE			
	1070	;UNA TECLA			
	1080				
A0B8 CD18BB	1090	CALL #BB18			
A0BB C9	1100	RET			
	1110				
	1120	;ROUTINA DE IMPRESION			
	1130				
A0BC CD75BB	1140	PRINT: CALL #BB75			
A0BF 1A	1150	BUC: LD A,(DE)			
A0C0 FEFF	1160	CP 255			
A0C2 C8	1170	RET 2			
A0C3 CD5ABB	1180	CALL #BB5A			
A0C6 13	1190	INC DE			
A0C7 18F6	1200	JR BUC			
	1210				
	1220	;TEXTOS A IMPRIMIR			
	1230				
A0C9 53494554	1240	TXT1: DEFB "SIETE"			
A0CE FF	1250	DEFB 255			
A0CF 56454E54	1260	TXT2: DEFB "VENTANAS"			
A0D7 FF	1270	DEFB 255			
A0D8 454E	1280	TXT3: DEFB "EN"			
A0DA FF	1290	DEFB 255			
A0DB 50414E54	1300	TXT4: DEFB "PANTALLA"			
A0E3 FF	1310	DEFB 255			





## PROGRAMA CARGADOR

```

10 FOR N=&A000 TO &A0E4
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<>&624F THEN PRINT "ERROR
  EN DATAS"
60 DATA 175,205,14,188,1,9,9
70 DATA 205,56,188,62,1,205,180
80 DATA 187,33,0,0,17,4,9
90 DATA 205,102,187,62,1,205,150
100 DATA 187,205,108,187,62,2,205
110 DATA 180,187,33,0,10,17,4
120 DATA 19,205,102,187,62,2,205
130 DATA 150,187,205,108,187,62,3
140 DATA 205,180,187,33,19,0,17
150 DATA 24,9,205,102,187,62,3
160 DATA 205,150,187,205,108,187,62
170 DATA 4,205,180,187,33,19,10
180 DATA 17,25,19,205,102,187,62
190 DATA 4,205,150,187,205,108,187
200 DATA 62,5,205,180,187,33,5
210 DATA 0,17,18,4,205,102,187
220 DATA 62,5,205,150,187,205,108
230 DATA 187,62,6,33,5,15,17
240 DATA 18,19,205,102,187,62,6
250 DATA 205,150,187,205,108,187,17
  5
260 DATA 205,180,187,62,7,205,144
270 DATA 187,33,9,8,17,201,160
280 DATA 205,188,160,33,11,7,17
290 DATA 207,160,205,188,160,33,13
300 DATA 9,17,216,160,205,188,160
310 DATA 33,15,7,17,219,160,205
320 DATA 188,160,205,24,187,201,205
330 DATA 117,187,26,254,255,200,205
340 DATA 90,187,19,24,246,83,73
350 DATA 69,84,69,255,86,69,78
360 DATA 84,65,78,65,83,255,69
370 DATA 78,255,80,65,78,84,65
380 DATA 76,76,65,255,0,0,0
  
```

ferencia a una rutina del firmware que se utiliza también para seleccionar los canales de impresión.

La función de dicha rutina es únicamente la de cambiar un canal por otro. Así pues, si hemos definidos dos ventanas en dos canales distintos, utilizando esta rutina se podrán intercambiar dichos canales.

El cambio de dichos canales lleva consigo el cambio de las plumas, del papel, de la posición del cursor, límites de la ventana, scroll de la ventana, estado del cursor, modo de escritura de texto y de gráficos, etc.

La rutina encargada de producir dicho efecto es la siguiente:

**INTERCAMBIO DE CANALES.**  
#BBB7

Produce el intercambio de dos canales de impresión. El canal actualmente seleccionado se mantiene constante.

**Condiciones de entrada.**

El registro B debe contener un número de canal y el registro C debe contener el otro.

**Condiciones de salida.**

Se pierde el contenido de los registros AF, BC, DE y HL. Los demás registros se preservan.

Dado que el número de canal no puede ser superior a 8, a la entrada en dicha rutina se produce el enmascaramiento de los números de canal para asegurarse de que no sean mayores a dicho valor.

# BASE REGULABLE



1.º Es adaptable a cualquier monitor, incluso TV de 26 pulgadas

2.º Gira hasta 360º



3.º Se puede inclinar hacia adelante y hacia atrás (así evitarás los reflejos)



4.º Protege tu mesa de los daños que pueda causar tu monitor



**Importado y distribuido por:**

**ENFA IBERICA, S.A.**

C/ Balandro, 39 bis, 2.º B. 28042 Madrid. Tel. 742 18 92, 742 91 51.



# SKY-REED

Programa realizado por **José María Martínez Arbex**

**El programa es un juego de guerra que consiste en la defensa de una posición en tierra bajo el bombardeo de aviones enemigos. Existen dos tipos de aviones: cazas y bombarderos (fácilmente diferenciables por su tamaño). Los cazas no representan ningún peligro para nuestras posiciones, puesto que no sueltan bombas. Derribarlos aumenta nuestra puntuación y nos acerca al objetivo fijado. Se considera cumplido el objetivo inicial cuando se han derribado 7 aviones enemigos. A medida que se cumple cada objetivo, la máquina lo aumenta en 3 más, lo cual hace que al final sea imposible sostener la posición.**



Para atacar a los aviones contamos con un bunker de antiaéreos con dos posibilidades de tiro; corto y largo alcance. Tenemos 15 obuses de largo alcance y 25 de corto alcance. Además contamos con una plataforma de misiles teledirigidos (5 misiles). Con estos medios hemos de ser capaces de contener la invasión. El manejo de nuestras armas es un poco complicado al principio, pero con unas cuantas partidas se domina por completo. Las teclas son:

SHIFT: tiro en corta distancia (bombardeo).

RETURN: lanzamiento de misil.

ENTER: tiro en larga distancia (recto).

SPACE: anula disparo efectuado y permite disparar de nuevo.

TECLAS DEL CURSOR: controlan el movimiento del misil.

A: eleva el cañón.

Z: baja el cañón.

Es preciso tener en cuenta que al derribar un avión, éste pueda caer sobre nuestras posiciones, causando daños irreparables. Si el bunker es destruido perdemos la posibilidad de disparar y si nos destruyen el camión lanzamisiles, ya no podremos emplearlos. También nuestros propios disparos (no anulados a tiempo), pueden destruirnos. Los daños disminuyen nuestra puntuación (lo cual se puede comprobar en la tabla final).

Los aviones salen uno por uno a

distinta altura y con distinta velocidad. Ambas (altura y velocidad) son tomadas en cuenta para calcular los puntos que se otorgan por derribarlo.

También el tiempo es importante. La batalla empieza a primera hora de la mañana (las horas se marcan en la parte inferior y se reconocen por el color del cielo) y debe determinarse la SKY-REED antes del amanecer del día siguiente, de lo contrario perdemos toda nuestra munición y, evidentemente, somos destruidos. Es necesario darse prisa pues al caer la noche, la visibilidad del avión se reduce considerablemente y las estrellas se confunden con nuestras balas.

Resumiendo, es un juego de puntería, precisión y cálculo en las caídas. Nota: mientras caen las bombas enemigas, no podemos disparar, ni tan siquiera mover el cañón. Con el misil hay que intentar dar al avión en la cabina del piloto y desde abajo, preferiblemente.



## TABLA DE VARIABLES

HORA	Marca la hora del día o de la noche
MUNICION1	Balas de largo alcance.
MUNICION2	Balas de corto alcance.
MIS	Número de misiles.
OBJETIVO	Número de aviones que es preciso derribar para cumplir el objetivo de detener la invasión.
AS	Contiene «D» si es de día y «N» si es de noche.
DESTR	Switch de bunker destruido o no (1 ó 0)
DESMIS	Switch de lanzamisiles destruido o no.
PUNTOS	Contiene la puntualización total.
DAÑOS	Contiene los daños que se sufren.
ALTURA	Altura del avión.
TIP	Tipo de avión (caza o bombardero) 1-2 caza 3 bomb.
VELO	Velocidad del avión.
ELEV	Elevación en grados del cañón.
AVDERR	Aviones derribados.
DESVIAC	Hace que los obuses de corto alcance sientan la fuerza de la gravedad.
BO	Indica al programa si está cayendo una bomba.
BOMX	Coordenadas de la bomba
BOMY	
MISIL	Indica al programa si se ha lanzado un misil.
XPO	Coordenadas del misil.
YPO	
PORC	
	Obtiene el porcentaje en función de los aviones derribados y el objetivo.
CAS	Calificación (malo, regular, bueno, etc.)
ESTR	Número de estrellas, de noche.
NUBE	Número de nubes, de día.
MIS	Contiene la forma del misil cuando cae (división o división entera).

El resto de las variables carecen de importancia o son temporales (bucles, contadores, etc.).

```

10 ' =====
20 ' =====
30 ' ===
40 ' ===
50 ' === SKY - REED ===
60 ' ===
70 ' === J.M. ARBEX ===
80 ' ===
90 ' === 1.986 ===
100 ' ===
110 ' ===
120 ' =====
130 ' =====
140 '
150 '
160 ' Definimos los simbolos grafic
os, e inicializamos algunas variabl
es.
170 '
180 '
190 CALL &BC02:HORA=7:municion1=15:
MUNICION2=25:mis=5:objetivo=7:DEG
200 SYMBOL AFTER 0:SYMBOL 37,0,0,2,
2,62,126,12:SYMBOL 33,133,66,57,166
,86,137,74,162:SYMBOL 64,15,15,15,1
5,15,255,255,255:SYMBOL 163,0,0,0,0
,33,74,218,255:SYMBOL 92,0,0,48,56,
28,8:SYMBOL 47,0,0,12,28,56,16:SYMB
OL 60,24,60,62,126,255,127,126,60
210 SYMBOL 129,0,0,0,192,192:SYMBOL

```



# Serie ORO

```

38,0,0,127,255,63,1:SYMBOL 39,6,14
,254,254,255,224,96:SYMBOL 123,128,
128,143,137,233,15,9,9:SYMBOL 125,2
24,128,143,137,233,15,9,9
226 MODE 0:WINDOW#2,1,40,1,15:WINDO
W#3,6,40,1,20:WINDOW#4,4,5,16,17:WI
NDOW#5,5,6,17,19:WINDOW#6,1,20,21,2
5:PAPER#6,9:CLS#6
230 PEN#6,5:LOCATE#6,2,2:PRINT#6,"M
UNICION AVIONES":LOCATE#6,10,3:PR
INT#6,"HOR."
240 '
250 ' Empieza el programa en si: co
mprobaciones iniciales de horario y
municion.
260 '
270 CLS#2:CLS#4:CLS#5:HORA=(HORA+1)
*ABS(HORA<24):fall=0:MOVE 0,0:MOVER
0,80:DRAWR 640,0,4:MOVER -5,-5
280 IF HORA>6 AND HORA<21 THEN A$="
0" ELSE A$="N"
290 IF A$="D" THEN INK 0,2:INK 1,3:
INK 2,26:INK 3,24:INK 4,1:INK 9,9:I
NK 6,6:BORDER 2
300 IF hora<10 OR hora>18 THEN INK
0,13:BORDER 13
310 IF hora=6 THEN municion1=0:muni
cion2=0:MIS=0
320 LOCATE#6,10,5:PEN#6,6:PRINT#6,H
ORA
330 '
340 ' Dibuja el canon.
350 '
360 '
370 ' Actualiza marcadores
380 PEN#6,5:LOCATE#6,1,3:PRINT#6,CH
R$(123):PEN#6,6:PRINT#6,municion1:
PEN#6,5:LOCATE#6,1,4:PRINT#6,CHR$(1
25):PEN#6,6:PRINT#6,municion2:PEN#
6,5:LOCATE#6,1,5:PRINT#6,"M":PEN#6
,6:PRINT#6,mis:LOCATE#6,16,3:PRINT#
6,avderr
390 IF (destr AND desmis)=1 OR (mun
icion1+municion2=0 AND desmis=1) OR
(destr=1 AND mis=0) OR (municion1+
municion2+mis=0) THEN GOSUB 400:GOT
O 1350 ELSE GOTO 440
400 SOUND 1,400,150,15:SOUND 1,500,
250,13
410 INK 0,25:INK 1,13:PEN 1:LOCATE
3,3:PRINT "I N V A S I O N":FOR p=
1 TO 1999:NEXT:CLS#2:ENT 2,20,-10,4

```



```

,20,20,3:FOR avion=1 TO 10:SOUND 1,
180,10,15,1,2,1
420 TAG:altura=INT(RND#200)+160:PLO
T 0,0,8:FOR x=640 TO -32 STEP -16:M
OVE x,altura:PRINT "% "":NEXT:NEXT
:GOTO 1350
430 TAGOFF:RETURN
440 tip=INT(RND#3)+1:altura=INT(RND
#200)+200:ph=620:velo=INT(RND#12)+1
450 IF A$="N" THEN INK 0,0:INK 1,4:
INK 5,26:GOSUB 480:BORDER 0:GOTO 49
0 ELSE GOSUB 460:GOTO 490
460 FOR NUBES=1 TO 6:a(nubes)=INT(R
ND#570)+50:b(nubes)=INT(RND#150)+20
0:NEXT
470 nubes=0:TAMA=INT(RND#10):FOR NU
=1 TO 6:PLOT 0,0,2:FOR Nube=1 TO TA

```

```

MA:TAG:MOVE a(nu),b(nu):MOVER 1,0:P
RINT "<":NEXT:TAGOFF:RETURN
480 FOR estr=1 TO 30:PLOT INT(RND#6
38)+1,INT(RND#235)+165,2:NEXT:RETU
R N
490 IF MUNICION1=15 AND MUNICION2=2
5 AND MIS=5 THEN GOSUB 510
500 GOTO 590
510 PEN 2:LOCATE 17,18:PRINT CHR$(1
38);STRING$(3,143):LOCATE 17,19:PRI
NT CHR$(138);STRING$(3,143):LOCATE
17,20:PRINT CHR$(138);STRING$(3,143
):PEN 1:LOCATE 17,17:PRINT CHR$(222
)+CHR$(127)+CHR$(127)+CHR$(127);
520 PEN 9:LOCATE 2,19:PRINT "<":PEN
1:LOCATE 2,20:PRINT "!":LOCATE 3,1
8:PRINT CHR$(143);CHR$(143):LOCATE
3,19:PRINT "a";CHR$(143):LOCATE 3,2
0:PRINT CHR$(143)+CHR$(143)
530 PEN 1:LOCATE 19,16:PRINT CHR$(1
36):LOCATE 12,19:PRINT "a":LOCATE 1
3,19:PRINT CHR$(143);CHR$(143):LOCA
TE 12,20:PRINT STRING$(3,131)
540 LOCATE 1,1:PRINT CHR$(22)+CHR$(
1):PEN 4:LOCATE 12,20:PRINT "0 0":P
EN 0:LOCATE 13,19:PRINT CHR$(129):P
EN 1:PRINT CHR$(22)+CHR$(0)
550 FOR VENT=0 TO 96 STEP 32:MOVE 5
35+VENT,110:DRAWR 0,12,1:DRAWR 12,0
:DRAWR 0,-12:DRAWR -12,0:NEXT
560 PRINT CHR$(22)+CHR$(1):PEN 9:LO
CATE 10,20:PRINT "<":LOCATE 4,20:P
RINT "<":LOCATE 18,17:PRINT "<":LOC
ATE 18,18:PRINT "<":LOCATE 18,19:PR
INT "<":PEN 1:LOCATE 18,20:PRINT "
"
570 LOCATE 1,1:PRINT CHR$(22)+CHR$(
1):PEN 11:LOCATE 1,20:PRINT CHR$(24
8):LOCATE 11,20:PRINT CHR$(249):LOC
ATE 12,20:PRINT CHR$(248):PEN 9:LOC
ATE 1,20:PRINT STRING$(20,"*");PEN
1:LOCATE 1,1:PRINT CHR$(22)+CHR$(0
):
580 RETURN
590 TAGOFF:elev=45:IF TEST(127,127)
>0 THEN destr=0:MOVE 128,128:DRAWR
COS(elev)*20,SIN(elev)*25,3 ELSE de
str=1
600 IF TEST(445,110)=0 THEN DESMIS=
1
610 IF desmis=0 AND mis>0 THEN LOCA
TE 1,1:PRINT CHR$(22)+CHR$(1):LOCA
TE 14,18:PRINT "\"+CHR$(8):PEN 7:P
RINT " ":PEN 1:LOCATE 1,1:PRINT CHR
$(22)+CHR$(1);
620 GOSUB 710
630 '
640 ' Deteccion del teclado.
650 '
660 IF TEST(445,110)=0 THEN DESMIS=
1
670 IF avderr>objetivo THEN GOTO 1
320
680 '
690 PRINT CHR$(22)+CHR$(1)
700 GOSUB 850:GOTO 730
710 RETURN:PAPER 8:LOCATE 1,23:PRI
NT "LARGO ALCANCE: ";municion1:LOC
ATE 1,24:PRINT "CORTO ALCANCE: ";mu
nicion2:PAPER 0:RETURN
720 IF A$="N" THEN GOSUB 480 ELSE G
OSUB 470:RETURN
730 IF destr=1 THEN 780
740 IF NOT INKEY(6) AND municion1>0
THEN MUNICION1=MUNICION1-1:desvia
=0:LAR=0:GOSUB 1010:GOSUB 710:CLS#2
:MOVE 128,128:DRAWR COS(elev)*25,SI
N(elev)*25,3:GOSUB 720
750 IF NOT INKEY(21) AND municion2>
0 THEN MUNICION2=MUNICION2-1:desvia
c=0:LAR=1:GOSUB 1010:GOSUB 710:CLS#
2:MOVE 128,128:DRAWR COS(elev)*25,S
IN(elev)*25,3:GOSUB 720
760 IF NOT INKEY(69) THEN CLS#4:CLS
#5:elev=elev+6*ABS(ELEV<174):MOVER
-2,2:DRAW 128,128,0:DRAWR COS(elev)
*25,SIN(elev)*25,3
770 IF NOT INKEY(71) THEN CLS#4:CLS

```



```

#5:elev=elev-6*ABS(ELEV>40):MOVER
1,0:DRAW 128,128,0:DRAWR COS(elev)*
25,SIN(elev)*25,3
780 IF desmis=1 THEN 800
790 IF NOT INKEY(18) AND mis>0 THEN
mis=mis-1:GOSUB 1230:IF fall=0 THE
N GOTO 270
800 '
810 ' Si no se pulsa tecla, mov. de
l avion
820 '
830 GOSUB 850:GOTO 950
840 '
850 TAG:PLOT 0,0,1:MOVE ph,altura:1
F tip=3 THEN PRINT "&";":SOUND
1,500,5,5,,1:ph=ph-3-velo:GOTO 870
860 PRINT "%":":SOUND 1,200,5,5,
,1:ph=ph-3-velo:ALTURA=ALTURA+INT(
RND#3)-1
870 IF (bo=1 OR INT(RND#100)>1) OR
tip<3 THEN 930
880 TAGOFF:PRINT CHR$(22)+CHR$(1):
TAG
890 bo=1:bomx=ph+5:bomy=altura-5:FO
R bomba=bomy TO 90 STEP -4:TAG:PLOT
0,0,0:MOVE bomx,bomy:PRINT ".":bo
mx=bomx-2:bomy=bomy-4:PLOT 0,0,1:MO
VE bomx,bomy:PRINT ".":SOUND 1,bom
ba,7,10
900 IF bomy>150 THEN 910 ELSE IF TE
STR(0,-5)<>0 OR TESTR(8,0)<>0 THEN
GOSUB 1210:bo=0:GOTO 930
910 IF INT(RND#10)>1 THEN 920 ELSE
GOSUB 840
920 NEXT
930 IF bo=1 THEN bo=0
940 TAGOFF:RETURN
950 IF PH>-16 THEN 730 ELSE CLS#2:G
OTO 270
960 '
970 ' Vuelve a comprobar tecla o mo
ver avion
980 ' Ha habido un disparo...
990 '
1000 '
1010 y=128+SIN(elev)*25:x=128+COS(e
lev)*25:TAG:PLOT 0,0,6:MOVE x,y+16:
PRINT "!":PLOT 0,0,3:MOVE x,y+16:P
RINT "!":MOVE x,y+16:PRINT " ":TA
GOFF:SOUND 1,300,8,15,,8:MOVE x,y:
w=COS(elev)*4:e=SIN(elev)*4:FOR dis
=1 TO 700 STEP 4:UNO=XPOS:DOS=YPOS
1020 desviac=desviac+0,03:IF INT(RN
D#10)<2 THEN bo=1:GOSUB 850:bo=0:MO
VE UNO,DOS:PLOT w,e-desviac*LAR,2
ELSE PLOT w,e-desviac*LAR,2
1030 '

```



```

1040 Comproueba si da al avion
1050 '
1060 '
1070 IF YPOS<90 THEN TAG:PLOTR -20,
10,4:PRINT "I";:PLOTR -25,5,3:PRINT
"#";:SOUND 1,200,10,15,,8:RETURN
1080 IF TESTR(1,2)=1 THEN TAG:PLOTR
-20,10,4:PRINT "I";:PLOTR -25,5,3:
PRINT "#";:SOUND 1,200,10,15,,8:IF
YPOS>180 THEN GOSUB 1170:GOTO 270
ELSE TAGOFF:LOCATE 5,5:PRINT " Cul
dado...":danos=danos+5:FOR pau=0 TO
999:NEXT:GOTO 1090
1090 PLOTR -1,-2,0
1100 IF NOT INKEY(47) THEN RETURN
1110 NEXT dis
1120 RETURN
1130 '
1140 ' Rutina de caida del avion
1150 '
1160 '
1170 IF tip=3 THEN MOVER -32,-5:PRI
NT " ";
1180 TAG:avderr=avderr+1:puntos=pun
tos+10+INT(altura/100)+INT(XPOS/100
)+velo:FOR caida=altura TO 92 STEP
-4:ph=ph-caida/65:SOUND 1,ABS(PH*1.
5),2,5:PLOT ph,caida,1:IF INT(RND*4
)>2 THEN PRINT "\"; ELSE PRINT "/";
1190 IF TESTR(0,-1)<>0 AND caida<15
0 THEN danos=danos+4
1200 NEXT:danos=danos+1:PLOTR -32,0
,3:PRINT "#";:PLOT 0,0,1:MOVE 0,-16

```



```

1210 MOVER -32,15:PRINT "I";:PLOTR
-32,0,3:PRINT "#";:FOR music=400 TO
390 STEP -1:SOUND 1,music,10,10,,
8:NEXT:RETURN
1220 '
1230 mi$="\":xpo=416:ypo=125:FOR MI
SIL=1 TO 20:SOUND 1,300+MISIL,2,15,
,,1:XPO=XPO-2:YPO=YPO+2:PLOT 0,0,6:
MOVE XPO,YPO:PRINT CHR$(5)+MI$;NEX
T:FOR misil=1 TO 100:MOVE xpo,ypo:P
RINT CHR$(5)+mi$;:ypo=ypo+3:IF ypo>
420 THEN FALL=1:RETURN
1240 IF ypo<100 OR TESTR(8,-14)<>0
THEN GOSUB 1210:PLOT 0,0,3:MOVE XPO
,YPO:PRINT "#";:TAGOFF:RETURN
1250 IF altura+4>ypo AND altura-4<y
po AND ph-9<xpo AND ph+9>xpo THEN T
AG:PLOTR -20,10,4:PRINT "I";:PLOTR

```

```

-25,5,3:PRINT "#";:SOUND 1,200,10,1
5,,8:TAGOFF:GOSUB 1170:RETURN
1260 IF NOT INKEY(8) THEN mi$="\":x
po=xpo-4
1270 IF NOT INKEY(1) THEN mi$="/":x
po=xpo+4
1280 IF NOT INKEY(0) THEN ypo=ypo+2
1290 IF NOT INKEY(2) THEN ypo=ypo-6
1300 IF RND>0.5 THEN bo=1:GOSUB 850
:bo=0
1310 NEXT misil:TAG:PLOT XPO-20,YPO
+5,3:PRINT "I";:PLOTR -20,5,6:SOUND
1,200,10,10,,5:PRINT "#";:danos=d
anos-(5*YPOS<130):SOUND 1,250,10,7,
,,6:TAGOFF:FALL=1:CLS#2:RETURN
1320 ' Si se ha conseguido detener
la invasion...
1330 '
1340 '
1350 SYMBOL AFTER 256:MODE 1:PAPER
0:CLS:PEN 0:CALL &BC02:BORDER 13:IN
K 0,26:INK 1,9:INK 2,0:INK 3,13:WIN
DOW 10,33,8,18:PAPER 1:CLS:MOVE 140
,106:DRAWR 390,0,1:DRAWR 0,184:DRAW
R -390,0:DRAWR 0,-184
1360 IF DEST=1 THEN DANOS=DANOS+10
1370 IF DESMI=1 THEN DANOS=DANOS+1
0
1380 porc=ROUND(avderr*100/objetivo
,1):LOCATE 2,2:PRINT "DESPERFECTOS:
";danos:LOCATE 2,4:PRINT "PUNTIUACI
ON: ";puntos:LOCATE 2,6:PRINT "PORC
ENTAJE: ";porc;%"
1390 LOCATE 2,8:PRINT "AVIONES DERR
IBADOS: ";avderr:IF porc<10 THEN ca$
="PESIMO"
1400 IF porc>=10 AND porc<20 THEN c
a$="MALO"
1410 IF porc>=20 AND porc<50 THEN c
a$="REGULAR"
1420 IF porc>=50 AND porc<70 THEN c
a$="NORMAL"
1430 IF porc>=70 AND porc<90 THEN c
a$="BUENO"
1440 IF porc>=90 THEN ca$="EXCELENTE
E"
1450 IF ca$="EXCELENTE" AND objetiv
o>20 THEN ca$="GENIAL"
1460 LOCATE 2,10:PRINT "NIVEL: ";ca
$;
1470 MOVE 0,0:DRAWR 639,0,2:DRAWR 0
,399:DRAWR -638,0:DRAWR 0,-399
1480 MOVE 0,0:DRAW 140,106,1:MOVE 6
40,0:DRAW 530,106:MOVE 640,400:DRAW
530,290:MOVE 0,400:DRAW 140,290
1490 WHILE INKEY<>"":c$=INKEY$:WEN
D:c$="":WHILE c$="":c$=INKEY$:WEND:
PAPER 0:IF avderr=objetivo THEN obj
etivo=objetivo+3:avderr=0:municion1
=15:municion2=25:mi$=5:hora=7:DESMI
S=0:DEST=0:GOTO 1500 ELSE RIU
1500 MODE 1:BORDER 0:PAPER 2:CALL &
BC02:CLS:LOCATE 2,24:PRINT "Nuevo 0
bjetivo: derribar";objetivo;"avione
s.":WHILE INKEY<>"":WEND:WHILE INK
EY$="":WEND:PAPER 0:GOTO 200

```



Compatible: CPC/464, CPC/664 y CPC/6128

# GANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANTAL

Porque pretendemos que **AMSTRAD SEMANTAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis los encargados de realizar estas páginas, en las que podréis aportar ideas y programas interesantes para otros lectores.

## Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANTAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.

- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:

- Descripción general del programa.
- Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.
- Instrucciones de manejo.
- Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.

- No se admitirán programas que contengan caracteres de control, debido a que no son correctamente interpretados por las impresoras.

- En una sola cinta puede introducirse más de un programa.

- Una vez publicado, **AMSTRAD SEMANTAL** abonará al autor del programa de 15.000 a 100.000 pesetas, en concepto de derechos de autor.

- Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.

- AMSTRAD SEMANTAL** se reserva el derecho de publicación o no del programa.

- Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANTAL**.

- Los programas sospechosos de plagio serán eliminados inmediatamente.

## ¡ENVIANOS TU PROGRAMA!

Adjuntando los siguientes datos:  
**Nombre y apellidos,**  
**dirección y teléfono.**

Indicando claramente en el sobre:

**AMSTRAD SEMANTAL**  
a **HOBBY PRESS, S. A. La Granja, 39**  
Pol. Ind. Alcobendas (Madrid)



# Ofites Informática

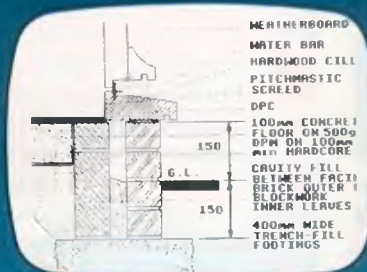
## Presenta: la tableta gráfica GRAFPAD II-

### LO ULTIMO EN DISPOSITIVOS DE ENTRADA DE GRAFICOS PARA AMSTRAD, COMMODORE Y BBC

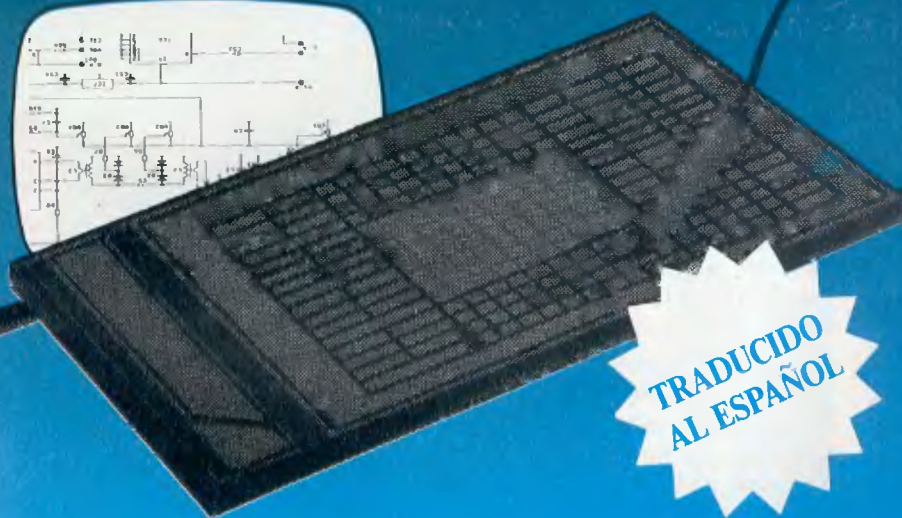
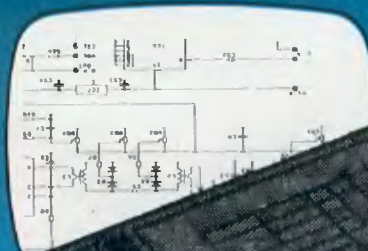
La primera tableta gráfica, de bajo costo, en ofrecer la duración y prestaciones requeridas por las aplicaciones de negocios, industria, hogar y educación. Es pequeña, exacta y segura. No necesita ajustes ni mantenimiento preventivo. GRAFPAD II es un producto único que pone la potencia de la tecnología moderna bajo el control del usuario.



DIBUJO A MANO ALZADA  
SOFTWARE DE ICONOS



DISEÑO DE ARQUITECTURA  
CON SOFTWARE DDX



TRADUCIDO  
AL ESPAÑOL

COMBINA EN UN UNICO DISPOSITIVO TODAS LAS PRESTACIONES DE LOS INTENTOS PREVIOS DE MECANISMOS DE ENTRADA DE GRAFICOS. LAS APLICACIONES SON MAS NUMEROSAS QUE EN LOS DEMAS DISPOSITIVOS COMUNES E INCLUYEN:

- selección de opciones
- entrada de modelos
- recogida de datos
- diseño lógico
- diseño de circuitos
- creación de imágenes
- almacenamiento de imágenes
- recuperación de imágenes
- diseño para construcción
- C.A.D. (diseño asistido por ordenador)
- ilustración de textos
- juegos
- diseño de muestras
- educación
- diseño PCB

#### ESPECIFICACIONES

RESOLUCION:

1.280 x 1.024 pixels.

PRECISION:

1 pixel.

TASA DE SALIDA:

2.000 pares de coordenadas por segundo.

INTERFACE:

paralelo.

ORIGEN:

borde superior izquierdo o seleccionable.

DIMENSIONES:

350 x 260 x 12 mm.

DISPONIBLE AMSTRAD:

CASSETTE ..... 23.900 ptas.

DISCO ..... 25.900 ptas.

(IVA NO INCLUIDO)

- FACIL DE USAR.
- TRAZADO PCB.
- C.A.D.
- AREA DE DISEÑO DIN A4.
- COLOR EN ALTA RESOLUCION.
- USO EN HOGAR Y NEGOCIOS.
- VARIEDAD DE PROGRAMAS DISPONIBLES.
- DIBUJO A MANO ALZADA.
- DIAGRAMAS DE CIRCUITOS.

DE VENTA EN LOS MEJORES COMERCIOS DE INFORMÁTICA

Si Vd. tiene alguna dificultad para obtener la tableta gráfica, puede dirigirse a:



Avda. Isabel II, 16 -8º

Tels. 455544 - 455533

Télex 36698

20011 SAN SEBASTIAN

CONDICIONES ESPECIALES PARA DISTRIBUIDORES



# ESTRUCTURAS DE DATOS (I)

## PILAS

*Basic en cuestión de estructuras está bastante desprovisto. La única estructura que puede utilizar Basic es la Matriz. Sin embargo, basta un poco de ingenio para que esta limitación del Basic se desvanezca ante nuestra imaginación.*



La **Matriz** es un conjunto de datos de la misma naturaleza, que se almacenan juntos. A esta estructura se le llama estática ya que no se altera, en el número de sus elementos, durante el transcurso del programa; ocupando todo el espacio de memoria que necesita, hasta que se elimine la misma.

En este artículo vamos a tratar las estructuras dinámicas de datos, es decir, aquellas que varían sus elementos, en la medida en que les hace falta. Estas estructuras son, a saber:

PILAS; LISTAS; COLAS Y ARBOLES

A cada una de estas estructuras dinámicas de datos, le dedicaremos un artículo completo.

Empezaremos por la más sencilla de todas: LA PILA

### ¿Qué es una pila?

Una pila, conocida también como lista directa, es una serie de datos que sólo pueden extraerse o introducirse por uno de sus extremos. Por ello se dice que una pila es una estructura del tipo último en entrar — primero en salir, **UEPSA** o **LIFO** en Inglés (last in, First out).

El extremo por el que se accede a la pila se le llama cima y al final de la misma se le conoce como base de la pila.

Las pilas sólo admiten dos operaciones, a saber:

SITUAR un elemento nuevo en la cima, a esta acción se le llama introducir o **apilar** un elemento (push,

para los que conocen el lenguaje ensamblador), y **SACAR** un elemento de la pila, que se conoce como extraer o vaciar la pila (pop en inglés).

Para trabajar con estas estructuras se utiliza una variable llamada puntero; que indica la posición de otra variable en la memoria del ordenador.

Basic no posee punteros y, por tanto, deberemos utilizar como medio de almacenamiento otra estructura con la que todos estamos familiarizados, la matriz o **array**, y utilizaremos como puntero una variable, que almacenará el valor del índice del último elemento introducido en la pila, el valor más alto de la matriz en el caso de que la pila se encuentre vacía.

### Cómo representar una pila en Basic

Según lo dicho anteriormente la única forma de trabajar con pilas en **Basic** es utilizar las matrices; almacenando todos los elementos de la estructura en la matriz. Por ello tenemos la primera limitación:

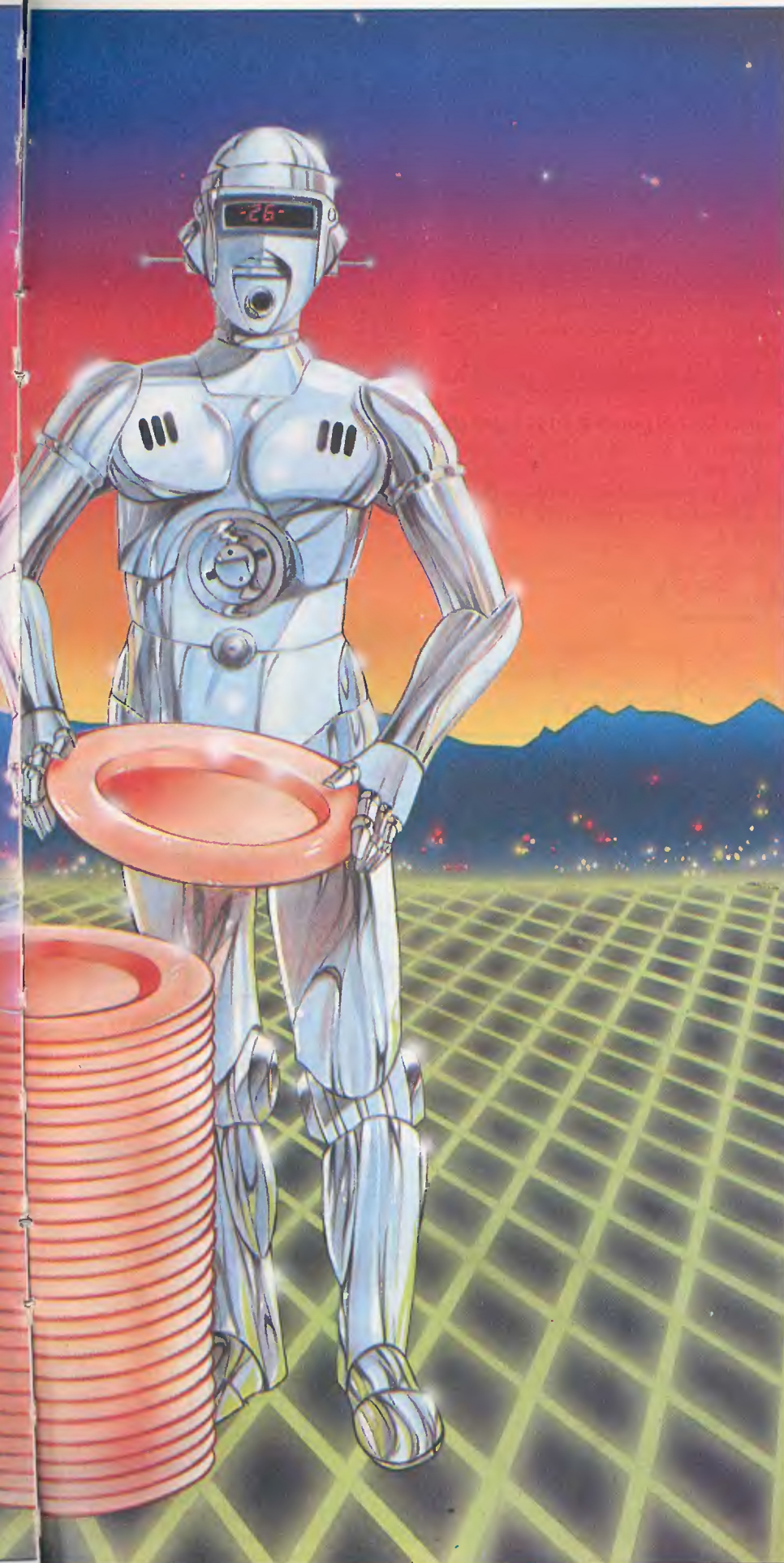
—La pila no puede hacerse nunca mayor que la matriz.

Normalmente y por convenio, se pone la base de la pila en el último elemento de la matriz, así conseguimos que esta estructura crezca «hacia arriba» dentro del array, y está llena cuando llegamos al primer elemento de la matriz.

El puntero de una pila equivale al índice del primer elemento libre del array, empezando siempre por el elemento mayor de la misma, de esta manera continuamente podremos determinar si la pila está vacía, o llena, ya que el puntero coincide con el último elemento de la matriz, o con el primero, si está llena.

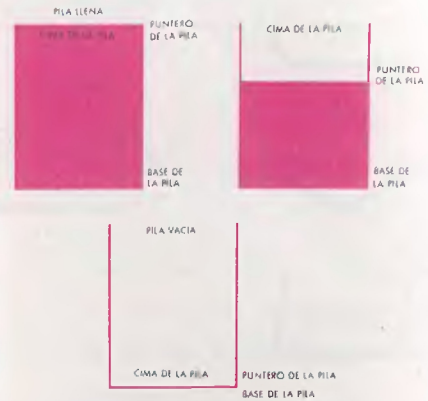






## PROGRAMACION

La figura 1 ilustra gráficamente el concepto de pila.



### Creación de una pila en Basic

Vamos a ver ahora una serie de módulos de programa, entiéndase subrutina, en los cuales vamos a simular, todos los procesos de manipulación de una pila.

Todo esto lo vamos a desglosar en una serie de pasos, lo que se conoce por algoritmo. En el programa ejemplo podemos ver todas las subrutinas, ya codificadas en Basic. El algoritmo general es el que sigue:

Crear una pila vacía.

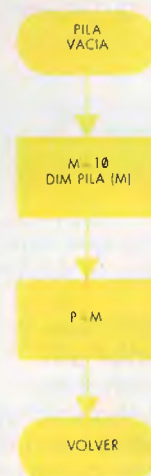
Introducir un elemento en la pila.

Extraer un elemento de la pila.

Vaciar e imprimir la pila.

Todo esto se presenta en el programa en forma de menú, para elegir cualquier acción y ver cómo se comporta esta estructura.

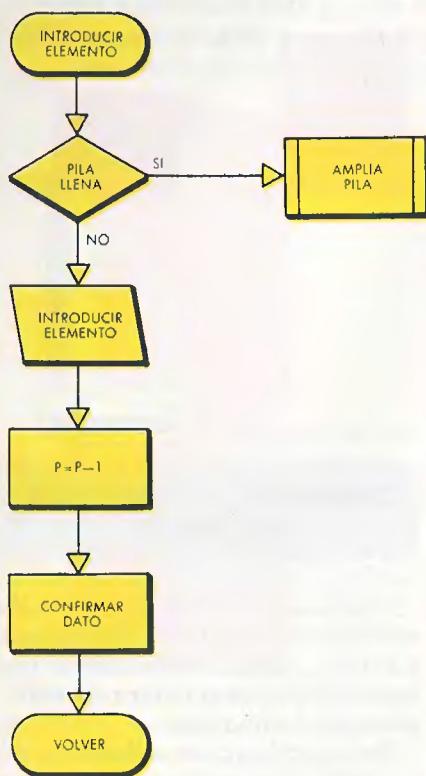
Veámoslo todo más desgornado.





## Creación de una pila vacía

Según lo explicado anteriormente, una pila está vacía, cuando el puntero de la misma indica el valor del



elemento mayor de la matriz. Por tanto, este módulo crea y sitúa el puntero de la pila, apuntando a la base de la misma.

Esta opción es condición indispensable para que el programa funcione correctamente.

El algoritmo es el siguiente:

Hacer  $M = 10$

Dimensionar la matriz PILA con M elementos.

Hacer el puntero (P) igual a M.

Volver.

El diagrama de flujo de la figura 2 representa gráficamente este algoritmo.

## Introducir un elemento en la pila

El algoritmo que indica el funcionamiento de esta subrutina es el siguiente:

Si la pila está llena ir a la subrutina de aumentar pila.

En caso contrario:

Insertar el elemento en el lugar indicado por el puntero de la pila.

Decrementar puntero.

Confirmar la introducción.  
Volver.

El diagrama de flujo correspondiente es el de la figura 3.

## Aumentar pilas

Crear una matriz con el mismo número de elementos que la pila.

Traspasar todos los elementos de la pila a la otra matriz.

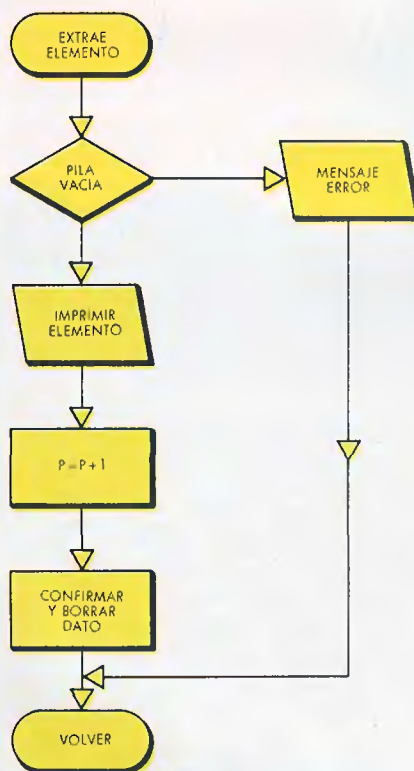
Borrar pila y dimensionar con diez elementos más.

Volver a situar todos los elementos en la pila.

Localizar el puntero en su lugar correcto.

Volver.

La representación gráfica de este algoritmo corresponde con la figura 4.



## Extraer un elemento de la pila

Este módulo funciona de la siguiente manera:

Si la pila está vacía indicar el fallo y volver.

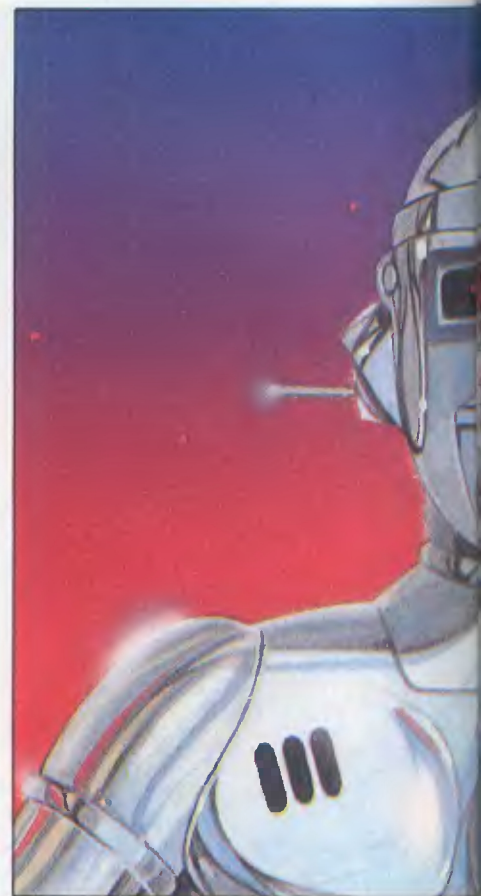
En caso contrario:

Imprimir el valor del elemento que indica el puntero.

Incrementar el puntero en uno.

Confirmar y volver.

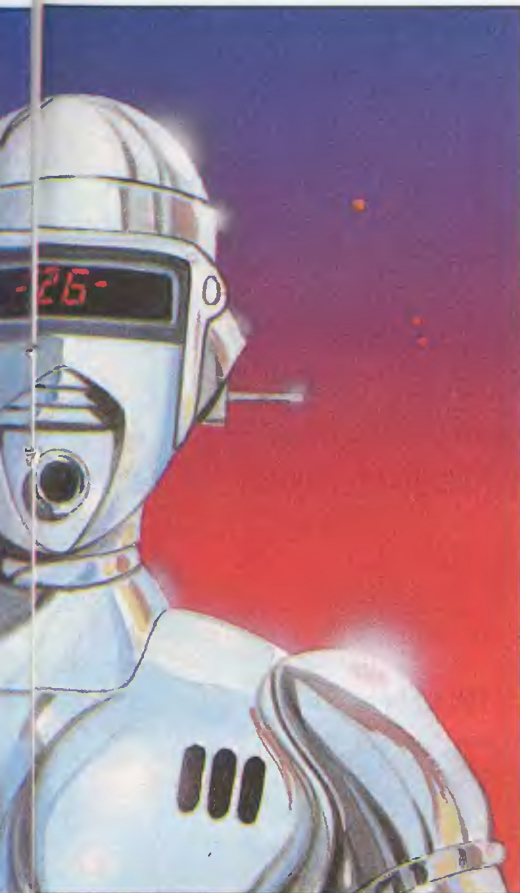
La pila está vacía si el puntero indica al último elemento de la matriz.



El diagrama de flujo correspondiente es el de la figura 5.







## Vaciar la pila

El algoritmo es muy simple y es el que sigue:

Repetir desde uno hasta el número máximo de elementos de la matriz:

Imprimir el elemento en curso de la matriz.

Poner a cero el mismo elemento.

Hacer el puntero igual al número máximo de elementos de la pila.

Confirmar y volver.

## Algunos usos de las pilas

Las pilas son muy usadas en informática. Principalmente por las subrutinas del sistema, la mayoría de los intérpretes de Basic utilizan las pilas para mantener el control de subrutinas, esto lo hacen de la siguiente forma:

Cuando se ejecuta un gosub los números de línea e instrucción son almacenados en la pila.

Si se ejecuta un **return** se extraen los dos elementos superiores de la pila, para devolver el control al punto donde se hizo la llamada, si la pila

## LISTA DE SUBROUTINAS

10-30	Comentarios
40-90	Menú de opciones
100-130	Control de subrutinas
140-190	Crear pila
200-280	Introducir un elemento
290-350	Extraer un elemento
260-480	Vaciar e imprimir pila
490-620	Ampliar pila
630-640	Pausa

está llena o vacía se produce el error.

En algunos ordenadores se utilizan



## LISTA DE VARIABLES

Pila	Matriz que representa la pila
Sec	Matriz para ampliar la pila
IS	Almacena el valor de INKEY
M	Número máximo de elementos de la pila
P	Puntero de la pila
Elem	Variable temporal que contiene el elemento a introducir en la pila
N	Variable de control

para el tratamiento de interrupciones y organización de la memoria, en este caso se utilizan uno o más registros para controlar la **pila**; estos registros actúan de punteros.

```

10 REM programa demostracion
20 REM del manejo de una pila
30 REM desde basic
40 CLS
50 LOCATE 7,2:PRINT "CREAR PILA";ST
  RING$(15,".");"1"
60 LOCATE 7,4:PRINT "INTRODUCTR UN
  ELEMENTO...2";
70 LOCATE 7,6:PRINT "EXTRAER UN ELE
  MENTO.....3";
80 LOCATE 7,8:PRINT "VACIAR PILA";S
  TRING$(14,".");"4"
90 LOCATE 14,22:PRINT "ELIGE OPCION
100 IS=UPPER$(INKEY$):IF IS="" THEN
  100
110 IF ASC(IS)<48 OR ASC(IS)>53 THE
  N 30
120 ON VAL (IS) GOSUB 140,200,290,36
  0
130 GOTO 30
140 REM CREA PILA
150 CLS
160 M=10:DIM Pila(M)
170 P=M
180 LOCATE 14,6:PRINT "PILA CREADA"
  :GOSUB 490
190 RETURN
200 REM INTRODUCIR UN ELEMENTO
210 CLS
220 IF P=M THEN LOCATE 6,6:PRINT "P
  ILA LLENA ESPERE UN MOMENTO":GOSUB
  490
230 LOCATE 9,6:INPUT "INTRODUCE UN
  ELEMENTO",ELEM
240 Pila(P)=ELEM
250 P=P+1
260 LOCATE 9,6:PRINT "EL ELEMENTO "
  ;P+1;" ES ";Pila(P+1)
270 GOSUB 630
280 RETURN
290 REM EXTRAER ELEMENTO
300 CLS
310 IF P=M THEN LOCATE 15,4:PRINT"P
  ILA VACIA":GOSUB 630:RETURN
320 LOCATE 10,6:PRINT "EL ELEMENTO
  ";P;" ES ";Pila(P)
330 Pila(P)=0:P=P-1
340 GOSUB 630
350 RETURN
360 REM VACTAR PILA
370 CLS
380 LOCATE 5,1:PRINT "ELEMENTOS QUE
  CONTIENE LA PILA";M
390 LOCATE 6,2:PRINT "PILA LLENA HA
  STA ELEMENTO ";ABS(P-M)
400 FOR N=M TO 1 STEP -1
410 PRINT Pila(N),
420 Pila(N)=0
430 NEXT
440 P=M
450 LOCATE 12,24:PRINT "PULSA UNA T
  ECLA"
460 IS=INKEY$:IF IS="" THEN 460
470 LOCATE 10,24:PRINT " PILA BORR
  ADA"
480 RETURN
490 REM AMPLIAR PILA
500 DIM SEC(M)
510 FOR N=1 TO M
520 SEC(N)=Pila(N)
530 NEXT
540 ERASE Pila
550 M=M+10
560 DIM Pila(M)
570 FOR N=M TO 10 STEP -1
580 Pila(N)=SEC(N-10)
590 NEXT
600 P=M-10:GOSUB 630:CLS
610 ERASE SEC
620 RETURN
630 FOR N=1 TO 2000:NEXT
640 RETURN
  
```



# TU PUEDES FORMAR PARTE DE



¡Como lo oyes! Tú puedes estar informado antes que nadie de todas las novedades que aparecen en el mercado, tener acceso a programas exclusivos y a precios especiales, poder conseguir los trucos que te ayuden a salir de esa pantalla que se te resiste, regalos, pósters, sorteos, pegatinas... y un montón más de ventajas.

## APUNTATE YA AL CLUB

No lo pienses más. Si te apuntas ahora al Club ERBE vas a recibir, además, como regalo de inscripción:

► El juego que más te guste de nuestro catálogo\*



► Una fabulosa camiseta con el anagrama del Club.



► Tu carnet personal de socio, que te dará derecho al 10% de descuento en todos los juegos de ERBE que compres, durante un año, en cualquiera de los establecimientos que aquí al lado se relacionan (o a través del Club si en tu zona no existiera ninguno).



CARNET VACILON  
PARA COMPRAR UN 10%  
MAS BARATO



JUEGO FLIPANTE



CAMISETA MOLONA

**TODO POR SOLO 3.000 PTS.... UN AUTENTICO CHOLLO**

Elige ya tu juego, corre al teléfono y llámanos al (91) 447 34 10 o escríbenos a ERBE SOFTWARE C/. Santa Engracia, 17 · 28010 Madrid para hacerte socio o pedir más información.

\* QUEDAN EXCLUIDOS LOS PROGRAMAS EN DISQUETE



## DISTRIBUIDORES AFILIADOS

### ALAVA

DATAVI. Avda. Gasteiz, 29. VITORIA

### ALMERIA

INFORMATICA-ELECTRONICA. Arapiles, 22.

### ALICANTE

INFOTRONICA. Doctor Jiménez Díaz, 2. ELCHE.  
MICRO CENTRO. César Ezequiel, 45. ALICANTE.  
MULTISYSTEM. San Vicente, 53. ALICANTE.  
CODE-2000. Ramón y Cajal, 3. DENIA.  
SILICON VALLEY. Glorieta, 4, semisótano. ELCHE (Alicante).

### ASTURIAS

CASA-RADIO MIERES. Jerónimo Ibrán, 11. MIERES.  
SOVI ELECTRONICA. Cabrales, 31. GIJÓN.  
BERNE. Menéndez Valdes, 13. GIJÓN.  
RADIO-NORTE. Uria, 20. OVIEDO.  
IMAGEN. Pablo Iglesias, 83. GIJÓN.  
COMERCIAL ARANGO. Marcos de Termiello, 2. AVILES.  
CUADRADO INFORMATICA. Torero, 5. OVIEDO.

### AVILA

DISCO-70. Plaza Sta. Teresa, 1.

### BADAJOS

SONIDO RUBIO. Avda. Fdez. Calzadilla, 10. BADAJOZ.  
RADIO GRAJERA. San José, 11. ALMENDRALEJO.

### BALEARES

ERGON. Falangista Laportilla, 2. P. MALLORCA.  
PROCONT. Extremadura, 31. IBIZA.  
COMPUSHOP. Via Alemania, 11. P. MALLORCA.  
TEKNOS. Aragon, 30. P. MALLORCA.

### CADIZ

PARODI-DISCOS. Novena, s/n.

### CASTELLON

CASIO, S. A. San Vicente, 6. Avda. Rey Don Jaime, 74.  
APARATOS. Mayor, 32. VINAROS.

### LA CORUÑA

PHOTOCOPY. Juana de Vega, 29-31.

### GERONA

CENTRAL FOTO. Ctra. de San Feliú, 28. PLATJA D'ARO.

### GRANADA

INFORMATICA-ELECTRONICA. Melchor Almagro, 8.

### GUADALAJARA

ABI. Padre Félix Flores, 3.

### GUIPUZCOA

SABA. Fuenterrabía, 14. SAN SEBASTIAN.  
AMASONIK. Paseo Colón, 80-82. IRUN.

### HUELVA

RADILUX. Concepción, 6.

### LEON

MICRO BIERZO. Carlos I, 2. PONFERRADA.

### LOGROÑO

COMPUTER PAPEL. Castroviejo, 19.

### LUGO

MED INFORMATICA. Avda. Ramón Ferreiro, s/n.

### MADRID

HIESA INFORMATICA. Camino V nateros, 40.  
INSTRUMENTOS MUSICALES ANGEL. Plaza España, 2 (local 9). LEGANES.  
COMPUTIQUE. Embajadores, 90.

### MALAGA

TODO INFORMATICA. Avda. Aurora, 14.  
INFORMATICA EUROPA. Moreno Carbonero (Edif. Carbonero). FUENGIROLA.  
ORGANIZACION EMPRESAS. Ricardo Soriano, 35. MARBELLA.  
TELEVISION PIÑAS. Dr. Eusebio Ramírez, 2. SAN PEDRO ALCANTARA.

### MURCIA

MEMORY SHOP. Lepanto, 1.

### NAVARRA

MICROORDENADORES RAMAR. Navarro Villoslada, 7.  
PAMPLONA.

### ORENSE

ALMACENES MENDEZ. Capitán Cortés, 17.

### PALENCIA

LA ESFERA. Mayor, 87.

### SANTANDER

RADIO MARTINEZ. Doctor Jiménez Díaz, 13.

### TOLEDO

CALCO. Angel de Alcázar, 56. TALAVERA DE LA REINA.

### VALENCIA

RADIO COLON. Colón, 7.  
ELECTRONICA MORANT. Jaime Torres, 12. GANDIA.

### VALLADOLID

CHIPS AND TIPS. Plaza Tenerife, 11.  
MICROLID. Gregorio Fernández, 6.

### VIZCAYA

REMBAT. General Concha, 12. BILBAO.  
EPROM 2. Juan XXIII, 3. SANTURCE.

### ZARAGOZA

ADA COMPUTER. Independencia, 24-26.

**FREE**

## ESCRITURA ITALICA

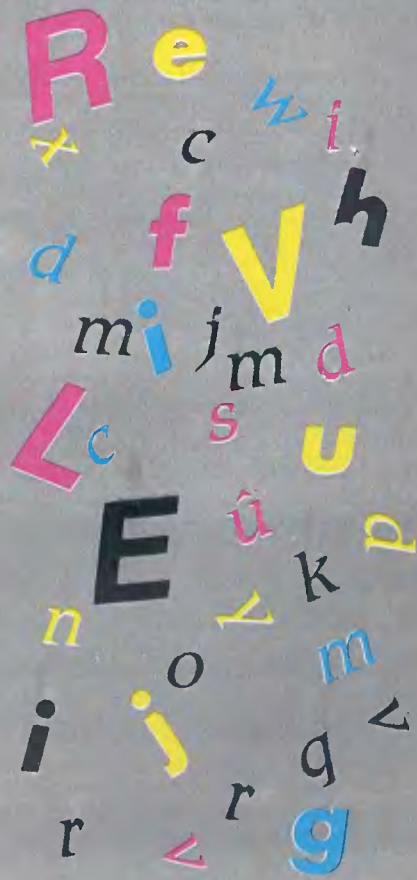
Juan Badía Font

**S**

e trata de una pequeña rutina que permite mejorar la presentación en pantalla de programas propios, ya que añade un nuevo tipo de letra: la itálica o inclinada. Asimismo permite también ver cómo quedará un determinado texto una vez pasado a impresora (si tiene este tipo de letra) antes de imprimirlo, etc.

Para introducirlo en tus programas tan sólo debes insertar las líneas 60-160, cambiando solamente las variables m, n de la línea 80 por las coordenadas en que deseas que aparezca el texto en itálica, y atribuir a la variable alfanumérica p\$ la palabra o palabras que deberán ser escritas en este tipo de letra.

El funcionamiento es sencillo: el programa escribe la palabra en el



borde inferior de la pantalla, y «mira» si cada punto está encendido o apagado. En caso de estar iluminado con un color de tinta diferente de O, imprime otro punto en el lugar de pantalla que hayamos indicado, sumándole un número creciente (x) para que se desplace a la derecha progresivamente, de los puntos inferiores a los superiores, y nos proporcione la letra inclinada. Las líneas 130-160 hacen solamente que el cursor se traslade al final de lo que hemos imprimido en itálica para poder seguir escribiendo a continuación en letra normal.

```
10 **** ITALIC ****
20 MODE 1
30 DIM L(255),M(0),N(0),O(0)
40 INPUT "palabra":p$
50 INPUT "Coordenadas (columna, línea)":L,N
60 PEN 2
70 LOCATE L,255:PRINT p$
80 PEN 1:LOCATE M,N
90 IF (PDG(N)-1)*16+L=400*(PDG(N)-1)+161
100 FOR V=0 TO 15 STEP 2
110 FOR W=0 TO LEN(p$)-1 STEP 2
120 IF TEST(W) THEN GOTO 140
130 NEXT W
140 FOR V=1 TO LEN(p$)-1
150 PRINT CHR$(V)
160 NEXT V
170 END
```



# Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

**SIN DUDA ALGUNA** está abierta a todos.

## PROGRAMAS PROTEGIDOS

Tengo dos cuestiones que haceros:

1.º Me gustaría saber cómo se puede salvar en una cinta un programa protegido. Por supuesto que no es para dedicarme a piratearlos (aunque no me creáis). Simplemente es que al ver vuestro especial con los Diez Mejores juegos, decidí comprarme el de «Fighting Warrior». Pues no llevaba ni dos días con él cuando se me enredó la cinta en el rodillo y por supuesto se fue a la perra. Os podéis imaginar cómo me quedé después de haber pagado 2.300 ptas. y no haber disfrutado del juego. Así pues, recurro a vosotros para que me digáis cómo copiar esos programas.

Por si os cabe alguna duda de mi honradez, os he de decir que ninguno de mis amigos tiene ordenador y de algunos conocidos que sí que tienen, no son Amstrad. Si no os parece ético publicar la manera de copiar un programa en el MICROHOBBY, pues entonces mandarme la solución a las señas que os adjunto.

2.º La segunda cuestión es: ¿cómo puedo conectar mi Amstrad a una cadena de sonido?

**Iñaki Erdoiza**

En primer lugar debemos decirte que por supuesto en ningún momento podemos dudar de la honradez de nuestros lectores. Respecto a la pregunta que nos formulas sobre la posibilidad de hacer una copia de un programa comercial, debemos decirte que dichos programas están muy bien protegidos, y que el único medio de poderlos copiar es deprotegerlos, cosa realmente complicada en muchos de dichos programas.

Respecto a tu segunda pregunta, únicamente debes conectar la clavija de tu cadena a la salida de estéreo que se encuentra en la parte lateral izquierda de tu ordenador.

## COLORES, MODOS, SONIDOS Y BASIC

Os escribo esta carta para que me aclaréis unas dudas sobre el Amstrad CPC 464:

¿Qué significa que el Amstrad sólo puede utilizar dos colores en modo 2 y cuatro colores en modo 1 y 16 en modo 0?

¿Cuántos colores puede tener dentro de un mismo carácter? ¿dos o más?

¿Es comparable el sonido del Amstrad con el del Commodore?

¿El Basic del Amstrad es más rápido que el del Spectrum?

Os agradecería que me respondierais claramente a la primera pregunta sobre todo.

**Lino González López**

Como todos sabemos, el Amstrad tiene distintos modos de resolución, en los cuales se pueden utilizar distinto número de colores. En el modo de más alta resolución (modo 2), únicamente están disponibles dos colores, en el siguiente modo (modo 1), tenemos cuatro colores y en el modo de menor resolución se disponen de 16 colores.

Dentro de un mismo carácter únicamente se puede obtener un color.

El sonido del Amstrad es tan bueno como lo pueda ser en el Commodore.

El Locomotive Basic es uno de los más potentes que existe en este momento.

## INCOMPATIBLES NO, GRACIAS

Poseo un CPC 464 y quisiera cambiarlo por el CPC 6128. El mayor problema que se me plantea es la posible incompatibilidad de los programas que actualmente poseo. Concretamente el que más me preocupa es el AMSWORD II.

Quisiera saber si lo puedo cargar en el 6128, ya sea a través de una copia en disco o por un cassette. Sea cual fuera la respuesta, quisiera que me orientaseis sobre un procesador de textos de la magnitud y calidad del AMSWORD II para el CPC 6128.

**Luis Gutiérrez Álvarez**

El AMSWORD II sí puede usarse con un 6128, pero, dado que se trata de un programa relativamente antiguo, no está diseñado para aprovechar al máximo el segundo banco de memoria del 6128. En nuestro país, Ofites Informática comercializa y distribuye un procesador de textos, el TASWORD II, que sí es capaz de utilizar dicho banco; esto te permitirá hasta 64 K de texto «on line» y una mayor velocidad de impresión.

## GENERACION DE CARACTERES

Soy un usuario de un Amstrad CPC 6128. El Amstrad, tiene caracteres gráficos que no son accesibles a través del teclado, como no sea con la instrucción CHR\$(número). Mi pregunta es:

¿Cómo se podría tener acceso a los demás caracteres desde el teclado sin tener que usar la instrucción CHR\$?

Ya que en un programa publicado en la revista MICROHOBBY AMSTRAD, se presenta el siguiente caso:

$FR = 440 * (20 + ((P - 10) / 12))$

El carácter "" no es accesible desde el teclado. Agradecería que contestaran lo más rápido posible.

**S. O. G.**

Los caracteres superiores al código ASCII, en general, no son accesibles desde teclado, por lo que la forma más fácil de obtenerlos es utilizando el comando CHR\$.

En cuanto al caso que nos comentas, suponemos se refiere al signo de exponenciación, el cual puede obtenerse pulsando simultáneamente las teclas MAYS + Pt.



# ESPEJO

# ANALISIS

**Nuevamente tratamos en esta sección las cadenas de caracteres. En esta ocasión analizamos un programa que mediante los comandos RIGHT\$ y LEFT\$ (y mucho ingenio) invierte las letras de la palabra que le digamos.**



El programa en sí es bastante sencillo. Una vez introducida la palabra se descompone en sus letras que son almacenadas en los elementos de una matriz, se analiza por los comandos anteriores y mediante GOTOs se invierte.

Empleamos lo que en programación se llama «**recursividad**» esto quiere decir que una de las rutinas que forman el programa se llama a sí misma. Lo que implica que hay que darle una condición de salida ya que si no estaría así indefinidamente.

Después se realiza el proceso de inversión y aparecen en pantalla la palabra primitiva y su inversa.

**10-50** Los REMs de siempre que nos indican el título.

**60** REM con el que comienza el programa principal.

**70** Se limpia la pantalla.

**80** Con INPUT se nos pregunta cuál es la palabra que queremos invertir. La expresión literal introducida es asignada a la variable «palabra\$».

**90-120** Llamadas a las distintas subrutinas creadas para realizar acciones concretas. Es-

tán ordenadas según sea necesaria su ejecución.

**130** Fin del programa principal. Esta instrucción sirve también para detener el mismo e impedir que la ejecución continúe por la primera subrutina.

**140** Comentario que nos indica dónde comienza la rutina que nos visualizará los resultados obtenidos.

**150-180** Son las encargadas de sacar en «bonito» todo lo que ha trabajado el programa. El texto lo escribe con tinta 1 (amarilla), la palabra a la que hemos dado la vuelta con tinta 3 (rojo brillante) y la invertida con tinta 2 (cyan intenso).

**190** Sirve para asignarle a la pluma el número de tinta que tenía en un principio (sólo para evitar sorpresas).

**200** Fin de la subrutina de «escritura de resultados». Devuelve el control al programa principal. Los RETURNs de las líneas 260, 330 y 380 actúan de un modo semejante con cada una de las rutinas donde están contenidos.

**210** Información, cómo no, del comienzo de una nueva subrutina: la de componer la palabra inversa.

**220** Comienzo del bucle WHILE... WEND que nos permitirá componer letra a letra el inverso de la palabra introducida.

**230** Comenzando por la última, almacenamos en «inverso\$» la concatenación de un

grupo de letras, ya invertidas, con la contenida en el elemento anterior de la matriz «letra\$». Es el indicado por «contador».

**240** Se decrementa el valor de «contador» para poder salir del bucle cuando sea igual a 0.

**250** Fin del mismo.

**270-280** REM indicativo de que ahí comienza la rutina de deletreo. Además nos informa que es recursiva, ya veremos por qué.

**290** Incrementamos la variable «contador» que, comenzando desde cero, nos va a servir para anotar el índice de cada una de las letras de nuestra palabra (o cantidad de veces que llamamos a la rutina recursivamente).

**300** Se define otra, «longitud», como el número total de caracteres de que consta la variable «imagen\$».

**310** Asignamos al elemento de índice «contador» de la matriz «letra\$» el carácter que ocupa el primer lugar a la izquierda de lo que quede del duplicado de la palabra introducida.

**320** Si la longitud de la palabra es distinta de 1, quitamos a «imagen\$» su primera letra, la ya almacenada, o la última empezando por la derecha. RIGHT\$ se encarga de hacerlo sin ningún problema.

Y llamamos de nuevo a la rutina de deletreo pero esta vez desde sí misma para repetir el proceso con el nuevo contenido de «imagen\$». Esto es la recursividad.

¿Cuándo terminará esta cadena de llamadas? En el momento en que la longitud de la imagen quede en un carácter. Entonces volveremos al programa principal.

**340** Comienzo de la rutina de inicializar datos.

**350** Dimensiona la matriz «letra\$» que tendrá un número de elementos igual a la longitud de la cadena literal (LEN (palabra\$)).

**360** Formamos una imagen auxiliar de la palabra para trabajar con ella sin perder la original.

**370** Inicializamos los valores de «contador» y de «inverso\$» que más tarde contendrá la palabra invertida.



**P**ara que tus dedos no realicen el trabajo duro, M. H. AMSTRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicítanoslo.



# Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A.**

**AMSTRAD SEMANAL.**

Apartado de correos 54.062  
28080 MADRID

**¡ABSTENERSE PIRATAS!**

**Vendo** los libros «Hacia la Inteligencia Artificial con **Amstrad**» y «Música y sonidos con **Amstrad**». También los siguientes juegos (versiones originales): Southern Belle; Dragontorc; Jump Jet; Beach Head; Alien 8; Exploding Fist; Decathlon; Rocky; Batalla de Midway;. Asimismo entrego cuatro cintas vírgenes C-15 y vendo todo el paquete por sólo 3.000 ptas. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

**Vendo** ZX Spectrum de 48K, impresora ZX Printer con papel, interface serie RS 232, interface joystick, microdrive con cartucho y varias cintas con juegos y utilidades. Se incluyen cables, accesorios y manuales completos. Está totalmente nuevo y se entrega embalado en sus cajas originales por solamente 25.000 ptas. URGE. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

**Vendo Amstrad CPC 464** con monitor F. verde. Incluyo manual en castellano y 8 programas originales. Seis meses de uso. Perfecto estado. Todo por 50.000 ptas. Tel. (96) 333 24 92. Preguntar por Guillermo.

**Vendo** videojuego Trenzi para TV blanco & negro y color. Dispone de tenis, hockey, frontón, squash, tiro al blanco y tiro al plato. Se entrega con los dos mandos para los jugadores, rifle desmontable, alimentador e instrucciones. Está en perfecto estado y lo vendo por tan sólo 5.000 ptas. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

**Vendo** ordenador de bolsillo & calculadora CASIO PB 100, con unidad central, teclado, pantalla y alimentación en una sola pieza. Se incluye un amplio manual de instrucciones y un libro de programación BASIC PB 100 (ambos en castellano), así como una funda protectora y dos pilas de litio CR 2032. Está absolutamente nueva y la vendo por tan sólo 4.000 ptas. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

**Vendo Amstrad CPC 6128** con monitor fósforo verde. Recién comprado con garantía oficial de AMSTRAD ESPAÑA. Adjunto programas de regalo (lúdicos y de utilidades). Todo por sólo 89.900 ptas. Interesados llamar al (972) 31 53 86. Preguntar por Miquel entre las 14 y 16 horas. O bien escribir a: Miquel Gasull. Alvaro de Bazán, 2. Palamós (Gerona).

**Desearía** cambiar el Everyone's a Wally (original), por uno de estos títulos: Dragonthor; Exploding fist; Pole Position; Southern Belle; Stars Trike; Knight Lore; Beach Head I o Beach Head II. También me interesarían otros títulos. Gerona, 30, 2.º San Adrià de Besós (Barcelona). Tel. (93) 381 08 36. Preguntar por Fernando.

**Vendo Amstrad CPC 6128** con 128K RAM, monitor de fósforo verde y unidad de disco 3" incorporada. Se incluye: Disco con CPM 2.2 y lenguaje; Logo; Disco con CPM Plus y utilidades; Disco con siete programas diversos; Seis discos vírgenes de 3"; Manual en castellano; Joystick Quickshot II; Fundas protectoras. Está completamente nuevo y se entrega perfectamente embalado por sólo 60.000 ptas. URGE. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

**Vendo Amstrad 464** (monitor color); en perfecto estado y con garantía. Regalo 30 juegos, un joystick, y 10 libros. Precio a convenir. Escribir a: Galicia, 15 apt.601. Las Palmas de Gran Canaria.

**Cambio** o vendo por **Amstrad CPC 464** fósforo verde o 50.000 ptas. por un Spectrum 48K dispuesto para su funcionamiento, con un cassette especial para ordenador más un interface tipo Kempston con salida de sonido por TV y su joystick correspondiente. Además todas las revistas de: Todospectrum, Microhobby, Micromanía, Input. También un libro de Basic, uno de código máquina, y uno de juegos. Incluyo en el lote 300 juegos educativos, recreativos y culturales cuyo valor es de 500.000 ptas. y también todos los copiones existentes: Mikie, Zorro, Rambo, Comando olé toro, Paradise, Alien 8, Rocky, Exploding fist, Contabilidad doméstica, Gens 3, Mons 3, Paintbox, Tans Spress, Baudcopy, Gmncopy, Editor, Beta Basic 3/0, Logo, etc. Llamar al tel. 375 28 81. Barcelona. Manuel Román Guerrero. Durazno, 1, 5.º puerta 4.ª Cornellá de Llobregat (Barcelona).

**Vendo** impresora SEIKOSHA gráfica GP-80M, Paralelo Centronics matriz 5x7; 30 cps. 80 caracteres por línea. Tracción. Ideal para SPECTRUM (comprando interface) para **Amstrad** (no trabaja en modo gráfico). Precio 18.000 ptas. Paco. Tel. (958) 60 52 95.

**Tengo** monitor fósforo verde GT65, y desearía cambiarlo por monitor color, pagando diferencia justa. Interesados llamar al teléfono 72 16 86 de Sevilla, y preguntar por Federico Noa Cruz.

## COMERCIAL LEVANTE BAZAR TETUAN

Distribuidor oficial de Amstrad, Spectravideo.

Dynadata, Toshiba, etc.

— AMSTRAD 6128 VERDE-DISCO	84.500 PTAS.
— AMSTRAD 6128 COLOR-DISCO	119.500 PTAS.
— AMSTRAD PCW 8256	129.000 PTAS.
— AMSTRAD 472 VERDE-CASSET	59.000 PTAS.
— AMSTRAD 472 COLOR-CASSET	90.000 PTAS.
— ORDENADOR DYNADATA	
— DPC 200-64 K	34.000 PTAS.
— IMPRESORA BROTHER M1009	36.000 PTAS.
— IMPRESORA BROTHER HR-S	21.000 PTAS.
— JOYSTICK INVESTIK	1.200 PTAS.
— JOYSTICK QUICK SHOT II	1.600 PTAS.
IVA INCLUIDO	

SERVICIO TECNICO

Arenal, 9. 28013 MADRID Tel.: 265 68 55



# En tu kiosco te espera algo muy inteligente

El AMSTRAD Especial número 2 incluye una cinta de cassette adherida a la portada con un lenguaje Lisp completo que

te permitirá comprender y dominar las técnicas más complejas de inteligencia artificial. Por si fuera poco, en nuestra

cinta se incluyen también dos concursos: uno, de diseño gráfico de pantallas, para artistas, te permitirá ganar hasta 170.000 pesetas en premios. En el segundo regalamos un ordenador Amstrad CPC6128.

El número 2 de AMSTRAD Especial trata un amplio espectro de interesantes temas, como un comparativo de impresoras, que le ayudará a elegir la más adecuada a sus necesidades, como multitud de programas y rutinas de utilidad en lenguaje máquina como un paquete de soft integrado, con tres programas en uno y un largo etcétera que sería demasiado prolijo detallar.

Si no lo encuentras en tu kiosco, solicítalo directamente a nuestra Editorial.

**MICROHOBBY**  
**AMSTRAD**  
*Especial* Año I N.º 2  
REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

475 ptas.

**SOFTWARE INTEGRADO**  
**3 PROGRAMAS EN UNO**

**TABLETA GRAFICA**  
**GRAFPAD II:**  
**EL ARTE POR ORDENADOR**  
**A TU ALCANCE**

**TE OFRECEMOS UN**  
**LENGUAJE DE**  
**PROGRAMACION LISP**  
**COMPLETO EN CINTA**  
**DE CASSETTE**

**NUESTRO PROGRAMA CREADOR**  
**DE CRUCIGRAMAS DESAFIA A TU INGENIO**

**IMPRESORAS:**  
**COMO HACER LA**  
**MEJOR ELECCION**

**ATENCION A NUESTRO**  
**FABULOSO CONCURSO:**  
**PUEDES GANAR UN**  
**CPC-6128 CON**  
**SOLO CARGAR LA CINTA**

Rellena este cupón y envíalo a **HOBBY PRESS, S. A.** Ap. de Correos 232. Alcobendas. Madrid.

Nombre \_\_\_\_\_ Apellidos \_\_\_\_\_  
Domicilio \_\_\_\_\_ C. Postal \_\_\_\_\_  
Localidad \_\_\_\_\_ Provincia \_\_\_\_\_  
Teléfono \_\_\_\_\_ Profesión \_\_\_\_\_ Fecha de nacimiento \_\_\_\_\_

¿Eres suscriptor de **MICROHOBBY AMSTRAD**? Sí ☐ No ☐  
Deseo recibir el Especial de **MICROHOBBY AMSTRAD** n.º 2 al precio de 475 ptas. (IVA incluido)

**FORMA DE PAGO**  
☐ Talón bancario adjunto a nombre de **HOBBY PRESS, S. A.**  
☐ Mediante Tarjeta VISA. N.º \_\_\_\_\_ Fecha de caducidad \_\_\_\_\_  
☐ Contra reembolso (supone 75 ptas. de gastos de envío).  
Fecha y firma: \_\_\_\_\_

A 01



# 4X1

**paga uno y llevate cuatro**

## **AMSTRAD**

COMBAT LINX  
GREMLINS  
DUMMY RUM  
DRAGONTORC  
MATCH DAY  
BASEBALL  
FIGHTING WARRIOR  
MAP GAME  
YIE AR KUNG-FU  
HYPERSPORTS  
ZORRO  
SUPERTEST  
PING-PONG

## **COMMODORE**

BASEBALL  
DROPZONE  
BEACH HEAD  
HYPERSPORTS  
SUPER ZAXXON  
FIGHTING WARRIOR  
SPY HUNTER  
TAPPER  
BC-II  
BOUNTY BOB  
POLE POSITION

## **MSX**

DISC WARRIOR  
JET SET WILLY II  
SHOWJUMPER

## **SPECTRUM**

ZAXXON  
FRANKIE  
BLUE MAX

BRUCE LEE  
RAID OVER MOSCOW  
BASEBALL  
DRAGONTORC  
ASTROCLONE  
GYROSCOPE  
MAP GAME  
ZORRO

COSMIC WARTOAD  
N.O.M.A.D.  
BATTLE OF PLANETS  
DYNAMITE DAN  
LEYENDA AMAZONAS  
BRIAN BLOODAXE  
PSYTRAXX

**.... y mil títulos más**

**!!! absolutamente  
originales!!!**

# 500

**ptas.**

***sinclair store***

**SOMOS PROFESIONALES**

BRAVO MURILLO, 2  
(Glorieta de Quevedo)  
Tel. 446 62 31 - 28015 MADRID  
Aparcamiento GRATUITO Magallanes, 1

DIEGO DE LEON, 25  
(Esq. Núñez de Balboa)  
Tel. 261 88 01 - 28006 MADRID  
Aparcamiento GRATUITO Núñez de Balboa, 114

AV. FELIPE II, 12  
(Metro Goya)  
Tel. 431 32 33 - 28009 MADRID  
Aparcamiento GRATUITO Av. Felipe II